MARVELL™

Timing Constraint Automation for Interface IP using Generative AI

Jeanne Trinko-Mechler
Fellow

Patricia Fong
Senior Staff Manager

Lukas Pettersson
STA Senior Engineer

Nicholas Hella
Software Staff Engineer

Roland Chen
Director AMS Design

11/7/2024 GSA WISH Conference

Agenda

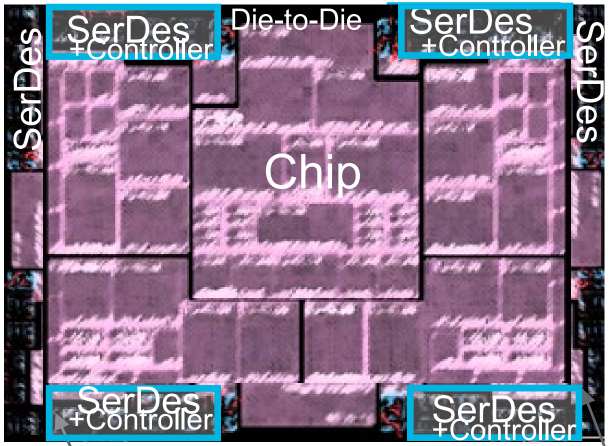
- Motivation for automating SDC constraints for interface IP using Generative AI
- Data Inputs to the Generative AI
- Framework and Architecture for Generative AI
- Demonstration of SDC Creator
- Conclusions and Next Steps

Motivation

All chips contain complex IP, such as SerDes, DDR, USB

IP require SDC (Synopsys Design Constraints) for implementation

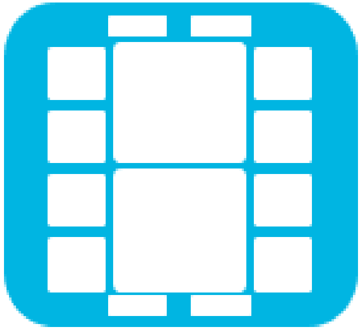
15%
Respins due to timing bugs



IO Interface Subsystems (SerDes, DDR, USB, Die-to-die…)

Challenges with manual SDC generation

- Hundreds lanes interface IP
- Multiple standards, configurable, >>parameters
- Many vendors
- Hundreds of pages: Specs
- Error-prone SDC Tcl code
- Chiplet architectures = more interfaces



SDC Timing Constraints

Dinner party analogy. Serve at the right time, delight guests.

1. **Courses (Signals):** Each course represents a signal (e.g., data, clock) in your design.
2. **Guests (Flip-Flops):** Your guests are the flip-flops (memory elements) waiting for their meal (signal).

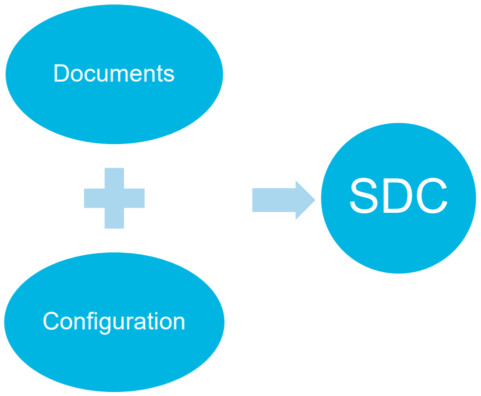
Timing Constraints (Serving Rules):

• **Set Input Delay:** Specify when waiter (signal) should arrive to guests (flip-flop).
Too early, guest won't be ready; too late, missed course.

• **Set Output Delay:** Ensures guest finishes course (output signal) before next course arrives.

SDC Creator Use Cases

1. IP Documents + Configuration → SDC

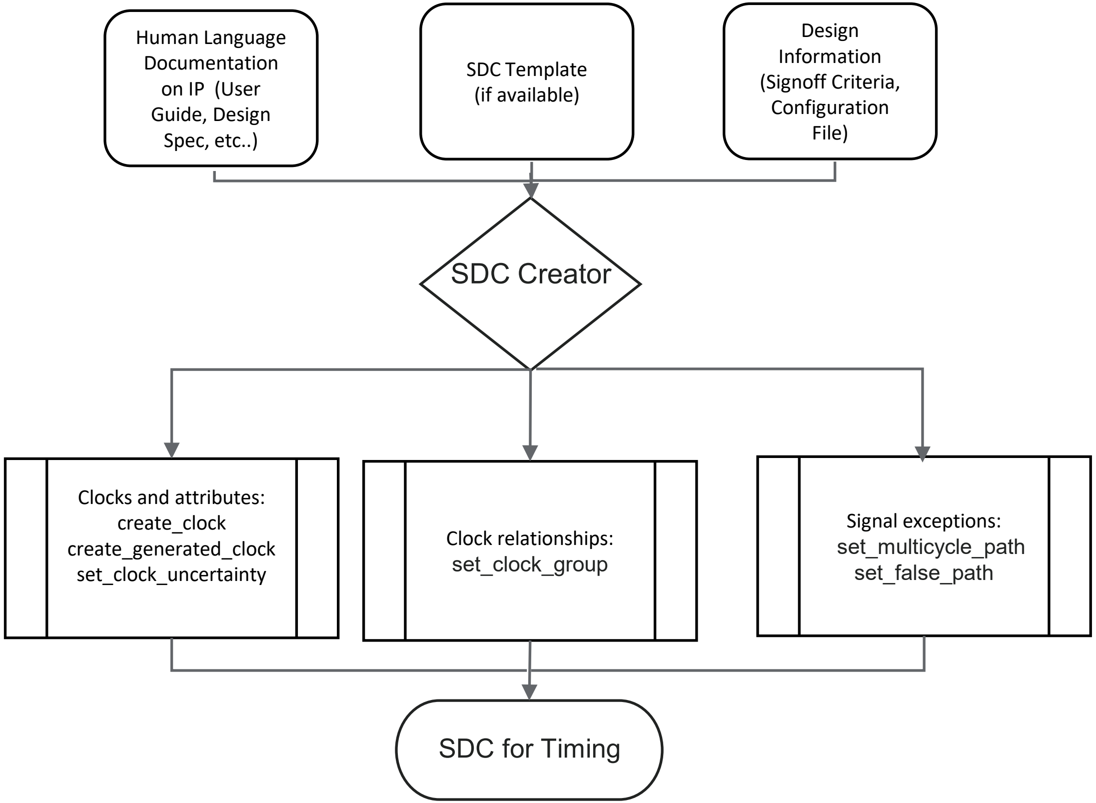


2. IP Documents + Generic Config → SDC Template

3. Complete SDC → Human Language format (Timing Constraint Review)

If Steps 1 & 2 result in incomplete SDC, improve IP documents

Automating SDC for a Design using Generative AI



Framework of SDC Commands

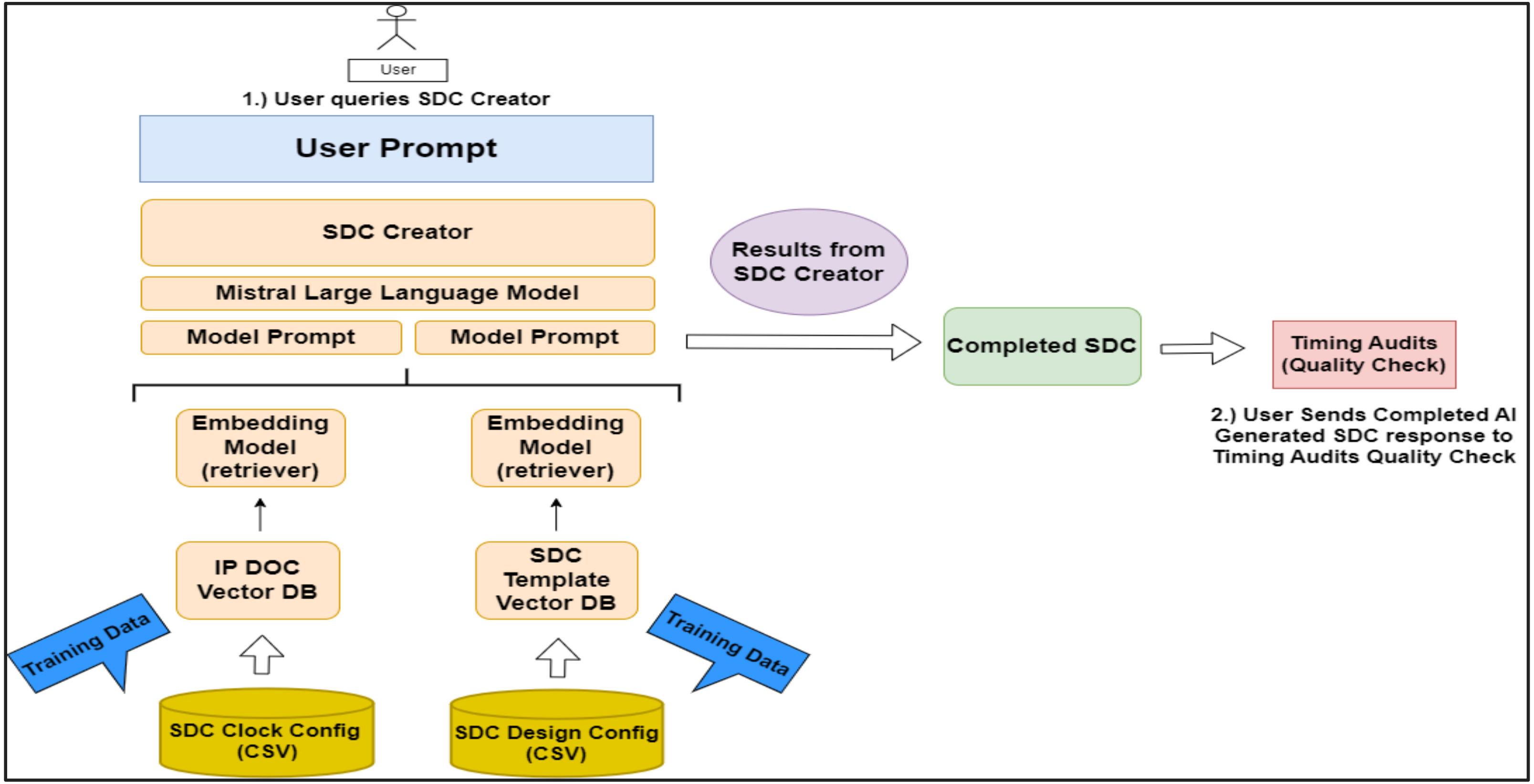
1. Create_clocks
    1. Command: create_clock create_generated_clock
    2. Design spec - Clock Config and Design Config
    3. Determined Mistral understood TCL and SDC

2. Clock uncertainty
    1. set_clock_uncertainty
    2. Design spec - Clock Config and Design Config

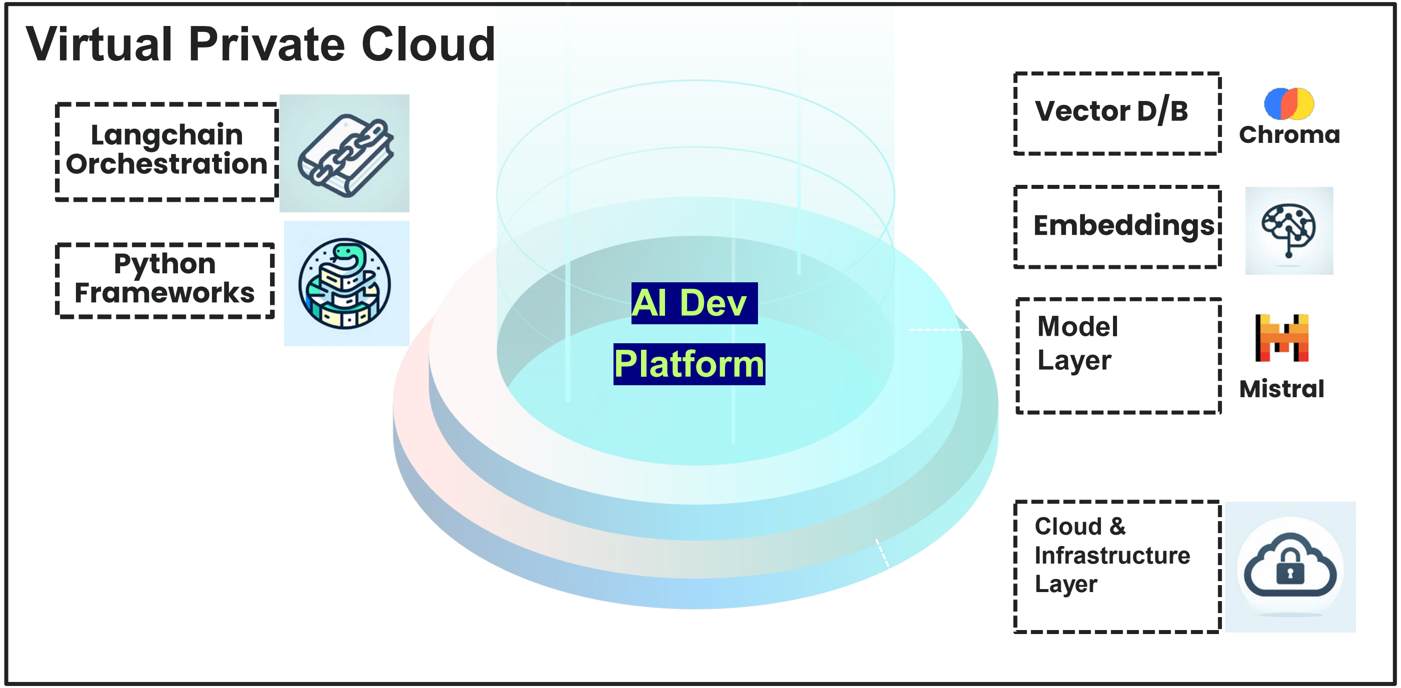
3. Clock grouping
    1. set_clock_group
    2. Design spec - Clock Config and Design Config
4. Exceptions
    1. set_max_delay, set_min_delay, set_false_path, set_multicycle_path …
    2. Design spec - Clock Config and Design Config
    3. SDC PCIE Templates

5. Skew checking - a file for post-timing processing
    1. set_skew
    2. Design spec - Clock Config and Design Config
    3. SDC PCIE Templates

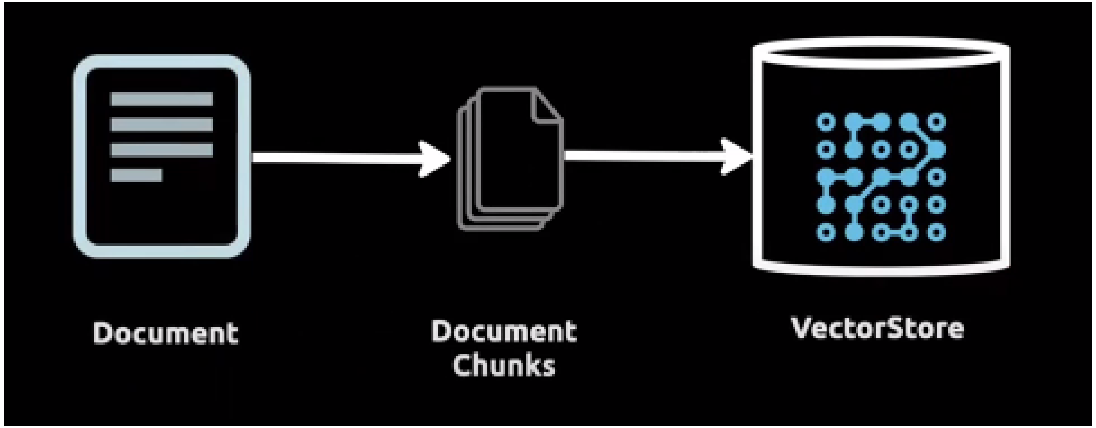
SDC Creator Usage Flowchart

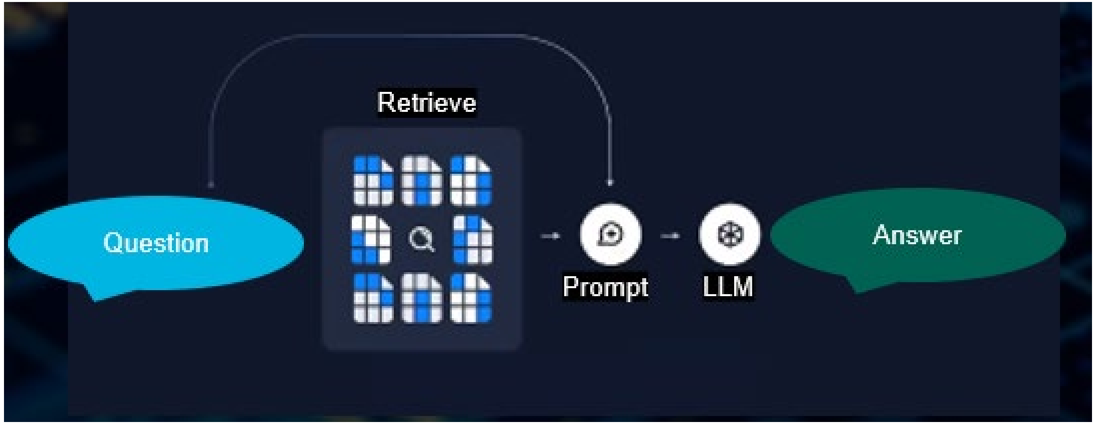


AI Platform Architecture for SDC Creator

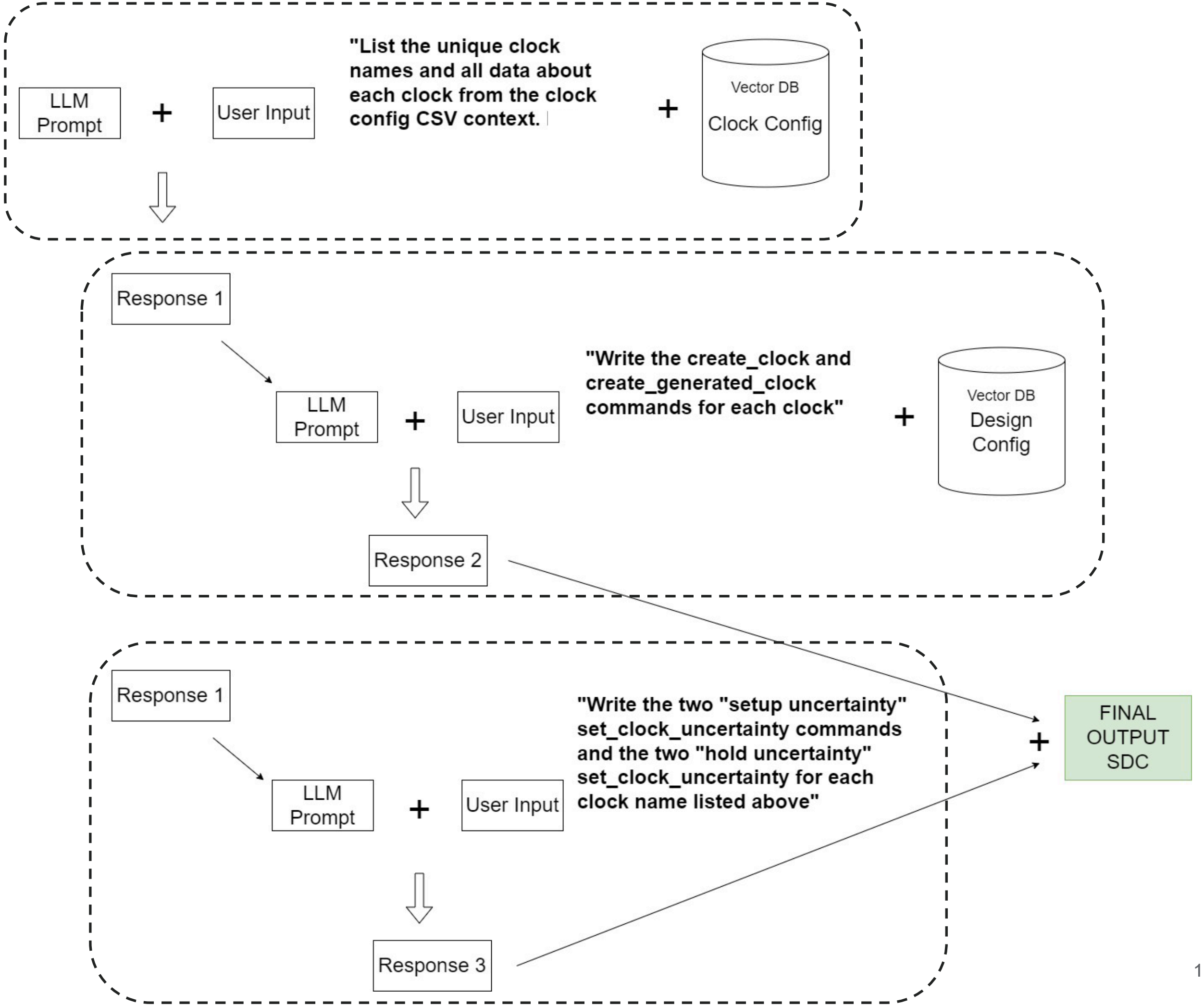


How SDC Creator uses RAG (Retrieval Augmented Generation)





SDC Creator Data Flow



Example LLM Prompt 1

- You are a Senior Timing Engineer responsible for writing the SDC (synopsys design constraints) code for IP.
- You know EVERYTHING about the TCL language and SDC (Synopsys Design Constraints) and always have correct syntax.
- Documentation will be provided to aid your answers and you must reference these documents in your reponse.
- You are working on a specific IP (intellectual property) called COMPHY, this comphy has two modes, PCIe and Ethernet, along with more configuration that the user may give you in their question.

Here are the rules for the response:

- IGNORE CLOCKS WITH 0 PERIOD VALUE PLEASE !

- Don't include references or justification for your response

Use this clock configuration CSV context to support your answer:

{sdc_clock_config_context}

Answer ONLY this question in your response:

{question}

Vector DB: Clock Config

Clocks	Period	Slow Uncertainty	Typical Uncertainty	Unit
PIN_CLK100M_125M_SIDE_A	10000	30	50	ps
PIN_CLK100M_125M_SIDE_B	10000	30	50	ps
AUX_CLK	2000	30	50	ps
PIPE_TXCLK_OUT	1000	30	50	ps
PIPE_SCLK_OUT	20000	30	50	ps
PIN_TXCLK_ALIGN_IN_REF	500	30	50	ps
PIN_REFCLKC_IN_SIDE_A_G1	0	0	0	ps
PIN_REFCLKC_IN_SIDE_B_G1	0	0	0	ps
PIN_REFCLKC_IN_SIDE_A_G2	0	0	0	ps
PIN_REFCLKC_IN_SIDE_B_G2	0	0	0	ps
PIPE_PCLK	0	0	0	ps
CORE_CLK	0	0	0	ps

Response 1: Clock Information

Unique Clock Names:
1. PIN_CLK100M_125M_SIDE_A
2. PIN_CLK100M_125M_SIDE_B
3. AUX_CLK
4. PIPE_TXCLK_OUT
5. PIPE_SCLK_OUT
6. PIN_TXCLK_ALIGN_IN_REF

Clock Details:

1. PIN_CLK100M_125M_SIDE_A
   - Period: 10000 ps
   - Slow Uncertainty: 30 ps
   - Typical Uncertainty: 50 ps

2. PIN_CLK100M_125M_SIDE_B
   - Period: 10000 ps
   - Slow Uncertainty: 30 ps
   - Typical Uncertainty: 50 ps

3. AUX_CLK
   - Period: 2000 ps
   - Slow Uncertainty: 30 ps
   - Typical Uncertainty: 50 ps

4. PIPE_TXCLK_OUT
   - Period: 1000 ps
   - Slow Uncertainty: 30 ps
   - Typical Uncertainty: 50 ps

5. PIPE_SCLK_OUT
   - Period: 20000 ps
   - Slow Uncertainty: 30 ps
   - Typical Uncertainty: 50 ps

6. PIN_TXCLK_ALIGN_IN_REF
   - Period: 500 ps
   - Slow Uncertainty: 30 ps
   - Typical Uncertainty: 50 ps

Successfully ignoring 0 period clocks, pulling in the correct data for each clock, and formatting the unit of measurement

LLM Prompt for Response 2

- If writing SDC code, each create_generated_clock command should minimally include -name, -cobinational, and -divide_by arguments with the correct values from the clock configuration CSV context.
- If writing SDC code, each create_generated_clock command should use a UNIQUE clock name. The -name argument should NOT have the same value across multiple create_generated_clock commands.
- If writing SDC code, use comments in your code to reference the context used to support your answer.
- If writing SDC code, all numbers must be converted to picoseconds if not already being used.
- Use the clock data to support your answer:

  {response}

- Use this design configuration CSV context to also support your answer:

  {sdc_design_config_context}

- Answer ONLY this question in your response:

  {question}

First LLM response propagates into the second prompt as means of chaining necessary data across multiple sources into context

Response 2 and Response 3 (excerpts)

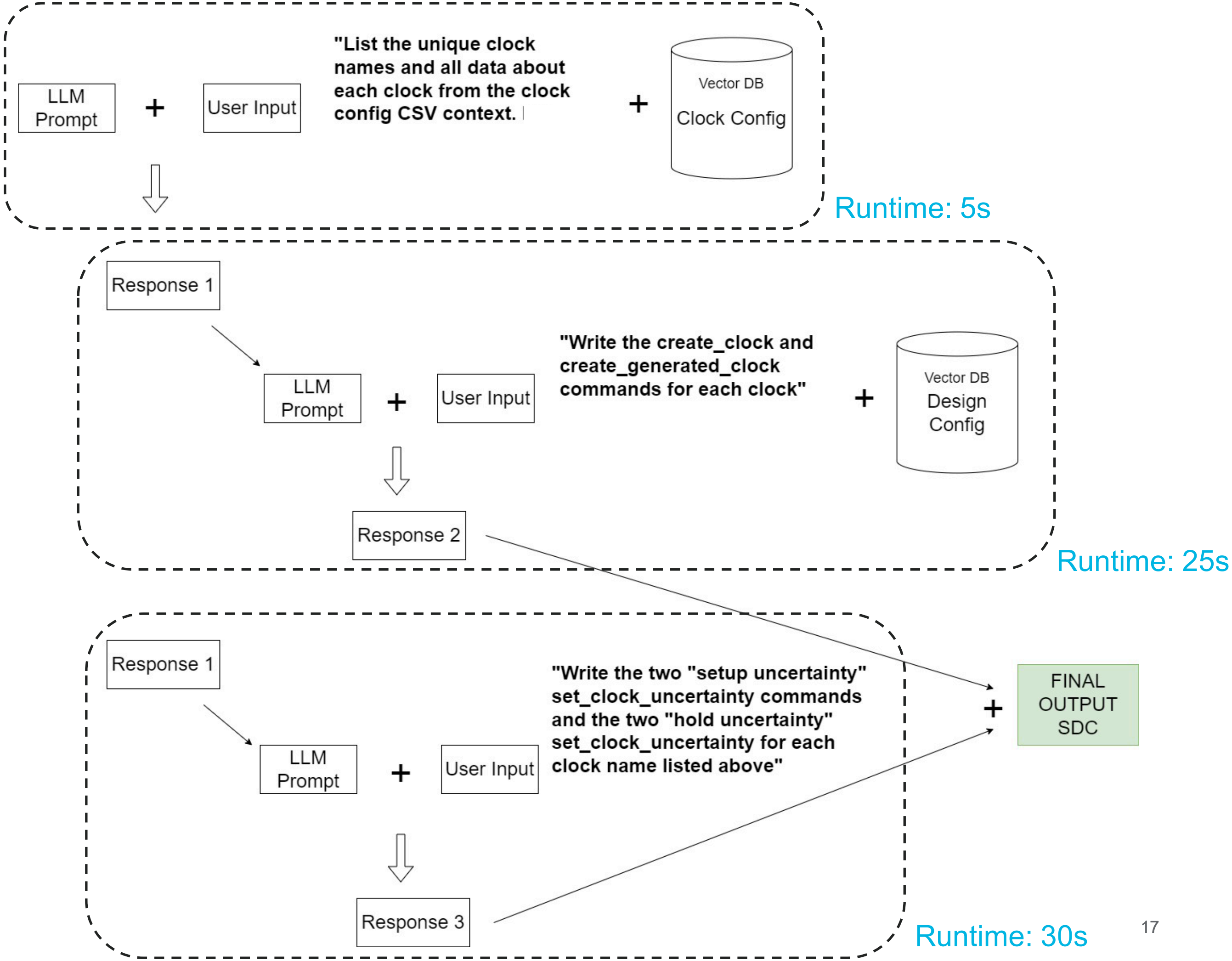
```
# PIPE_TXCLK_OUT
create_clock -name PIPE_TXCLK_OUT -period 1000 -waveform {0 500} [get_pins u_chipname_core/u_chipname_slice_host_i0/u_chipname_slice_host_i/u_fb_nic_pcie_ss/u_pcie_subsys_wrap/u_comphy_wrap/u_WRAP_COMPHY_56G_PIPE5_X4_4PLL/HC/PIPE_TXCLK_OUT]
create_generated_clock -name PIPE_TXCLK_OUT_GEN -divide_by 1 -source [get_pins u_chipname_core/u_chipname_slice_host_i0/u_chipname_slice_host_i/u_fb_nic_pcie_ss/u_pcie_subsys_wrap/u_comphy_wrap/u_WRAP_COMPHY_56G_PIPE5_X4_4PLL/HC/PIPE_TXCLK_OUT]

# PIPE_SCLK_OUT
create_clock -name PIPE_SCLK_OUT -period 20000 -waveform {0 10000} [get_pins u_chipname_core/u_chipname_slice_host_i0/u_chipname_slice_host_i/u_fb_nic_pcie_ss/u_pcie_subsys_wrap/u_comphy_wrap/u_WRAP_COMPHY_56G_PIPE5_X4_4PLL/HC/PIPE_SCLK_OUT]
create_generated_clock -name PIPE_SCLK_OUT_GEN -divide_by 1 -source [get_pins u_chipname_core/u_chipname_slice_host_i0/u_chipname_slice_host_i/u_fb_nic_pcie_ss/u_pcie_subsys_wrap/u_comphy_wrap/u_WRAP_COMPHY_56G_PIPE5_X4_4PLL/HC/PIPE_SCLK_OUT]
```

```
# PIPE_TXCLK_OUT
# Period: 1000 ps
# Slow Uncertainty: 30 ps
# Typical Uncertainty: 50 ps
set_clock_uncertainty -setup 30 -fall_from [get_clocks PIPE_TXCLK_OUT] -rise_to [get_clocks PIPE_TXCLK_OUT]
set_clock_uncertainty -setup 50 -rise_from [get_clocks PIPE_TXCLK_OUT] -fall_to [get_clocks PIPE_TXCLK_OUT]
set_clock_uncertainty -hold 30 -fall_from [get_clocks PIPE_TXCLK_OUT] -rise_to [get_clocks PIPE_TXCLK_OUT]
set_clock_uncertainty -hold 50 -rise_from [get_clocks PIPE_TXCLK_OUT] -fall_to [get_clocks PIPE_TXCLK_OUT]

# PIPE_SCLK_OUT
# Period: 20000 ps
# Slow Uncertainty: 30 ps
# Typical Uncertainty: 50 ps
set_clock_uncertainty -setup 30 -fall_from [get_clocks PIPE_SCLK_OUT] -rise_to [get_clocks PIPE_SCLK_OUT]
set_clock_uncertainty -setup 50 -rise_from [get_clocks PIPE_SCLK_OUT] -fall_to [get_clocks PIPE_SCLK_OUT]
set_clock_uncertainty -hold 30 -fall_from [get_clocks PIPE_SCLK_OUT] -rise_to [get_clocks PIPE_SCLK_OUT]
set_clock_uncertainty -hold 50 -rise_from [get_clocks PIPE_SCLK_OUT] -fall_to [get_clocks PIPE_SCLK_OUT]
```

SDC Creator Data Flow

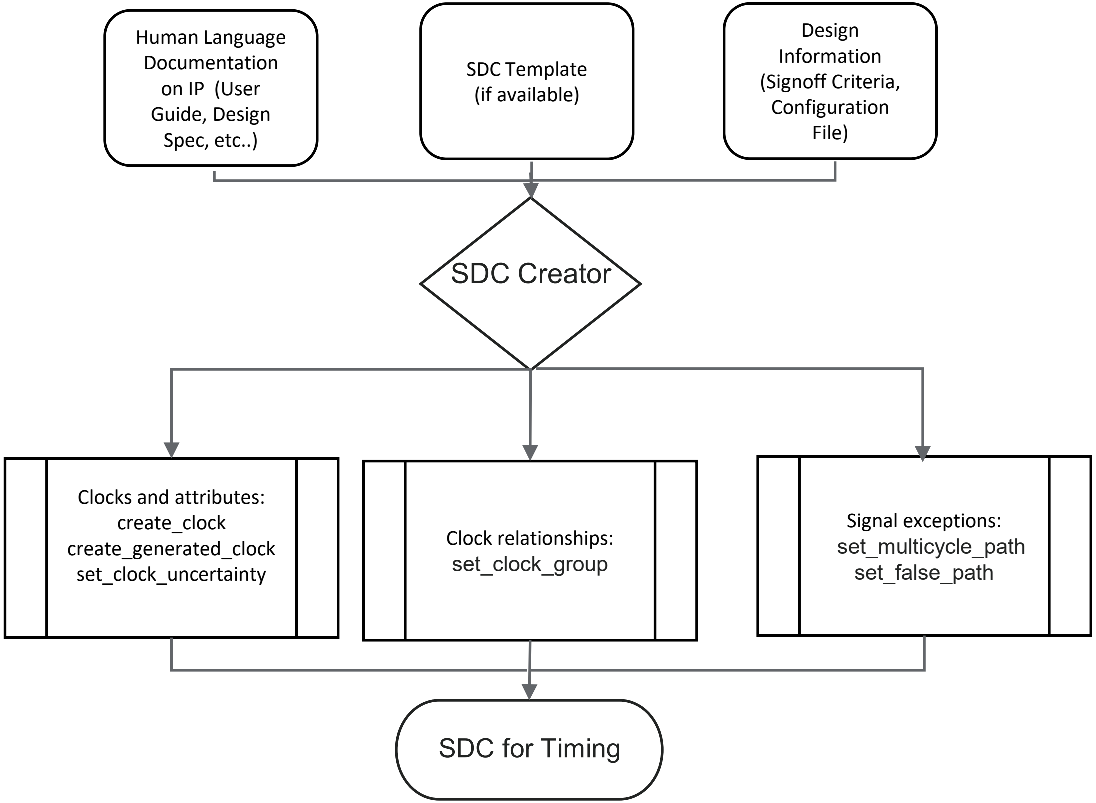


Runtime Statistics

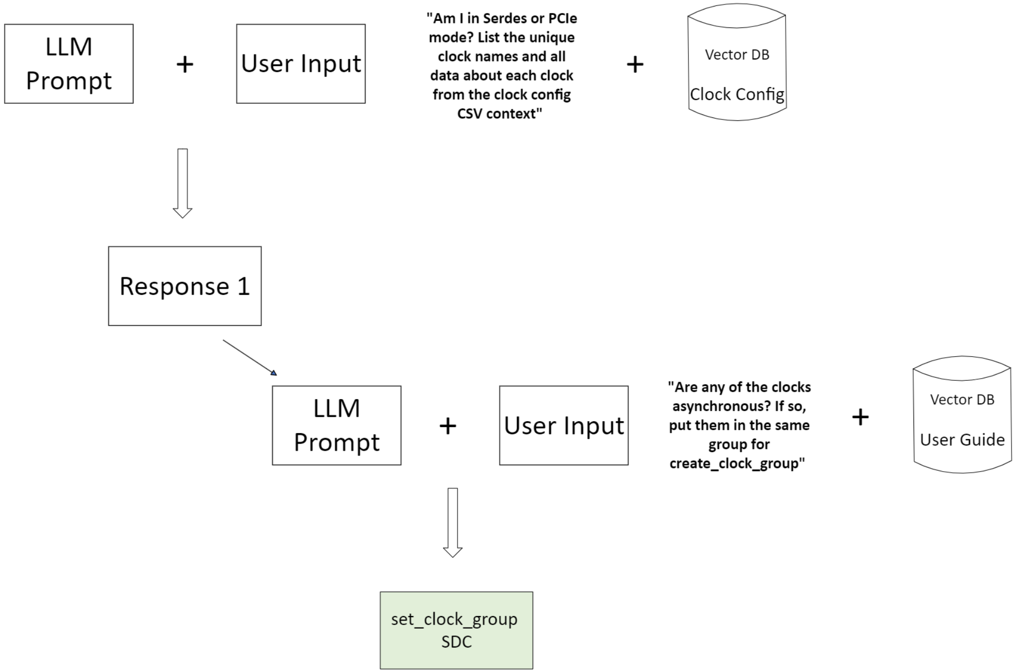
Action	Time
Response 1: Clock Context	5 seconds
Response 2: Clock Creation SDC	25 seconds
Response 3: Clock Uncertainty SDC	30 seconds
Per SerDes Instance	1 minute

Dramatically faster than hours needed to hand code SDCs

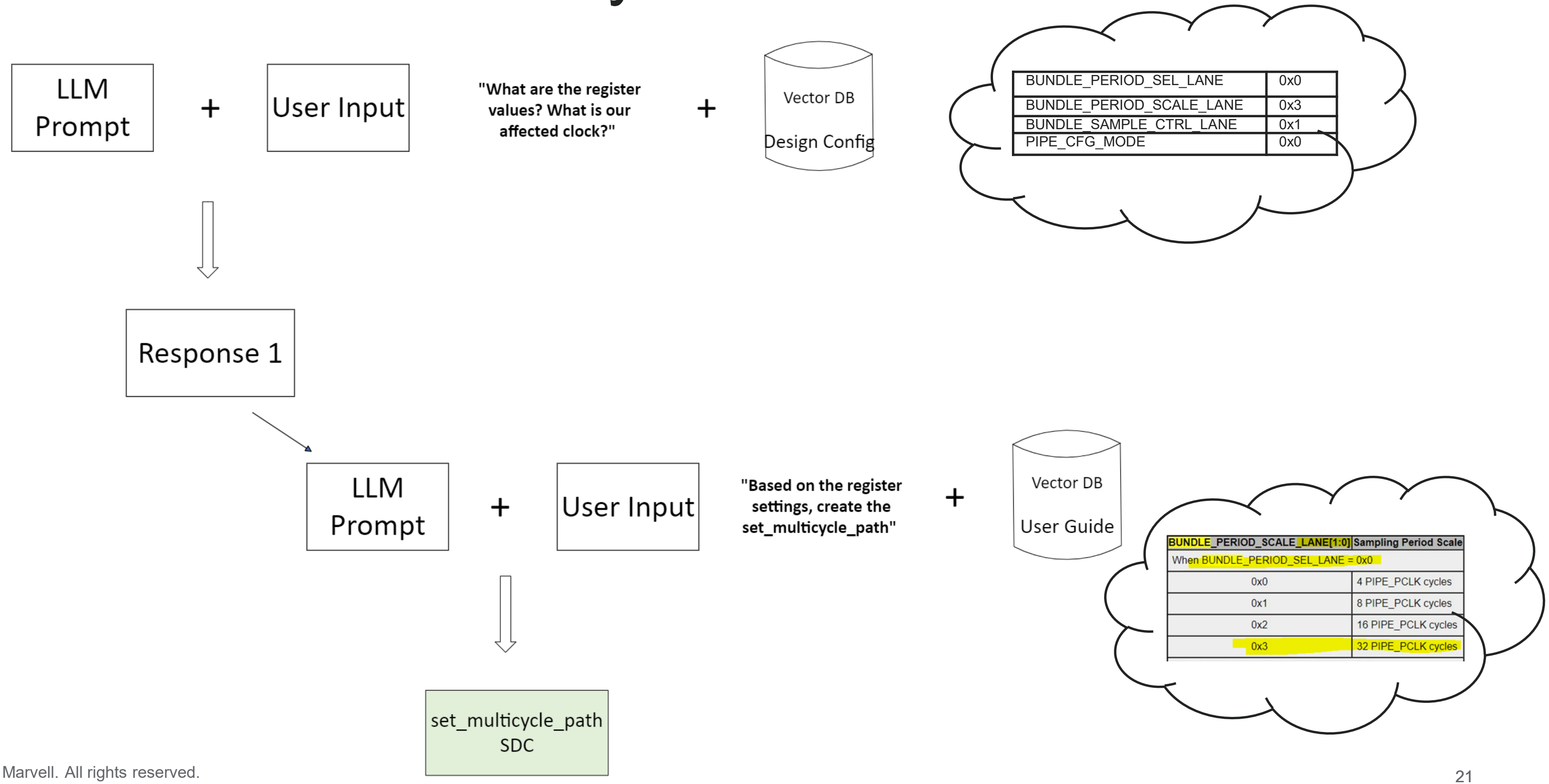
Automating SDC for a Design using Generative AI



SDC Creator for Clock Grouping



SDC Creator for Multicycle Paths



SDC to Human Language for Constraint Validation

01

Input a complete interface SDC

02

Translate commands and comments into human language.

Extract configurations

03

Compare output config to input config

Compare output human language to IP documentation and specifications

SDC to Human Language Example:

- ## Question:

From the SDC, provide information about CM0_CLK_RATE1_O

- ## Answer:

Here is information about CM0_CLK_RATE1_O from the SDC:

- **Definition**: CM0_CLK_RATE1_O is an output clock pin of the PHY that corresponds to the internal pin AFE_CM0_CLK_PLL[1][2]. It is one of the important PHY output clock domains where data may be timed synchronously[3].
- **Frequency**: The frequency of CM0_CLK_RATE1_O is determined by the parameter freq_rate1, which is set according to the design specification and the latest databook. For example, if the data rate is 16 Gbps and the data width is 32 bits, then freq_rate1 = 500 MHz[4].
- **Waveform**: The waveform of CM0_CLK_RATE1_O is defined by its period and duty cycle. The period is the inverse of the frequency, and the duty cycle is the fraction of the period that the clock signal is high. The minimum and maximum duty cycle values are 0.40 and 0.60, respectively, as specified in the databook.
- **Uncertainty**: The uncertainty of CM0_CLK_RATE1_O is the amount of variation in the arrival time of the clock signal due to various factors such as jitter, setup and hold margin, and duty cycle distortion. The setup and hold uncertainties are calculated by adding the pure margin, the clock jitter, and the duty cycle uncertainty. The clock jitter is 0.065 ns, and the duty cycle uncertainty is half of the difference between the maximum and minimum duty cycle values.

Summary

1. IP Documents + Configuration → SDC Creator → SDC
2. IP Documents + Generic Configurations → SDC Creator → SDC Template
3. Complete SDC → SDC Creator → Human Language / Constraint Review

Thank You

MARVELL™

Essential technology, done right™

Q&A