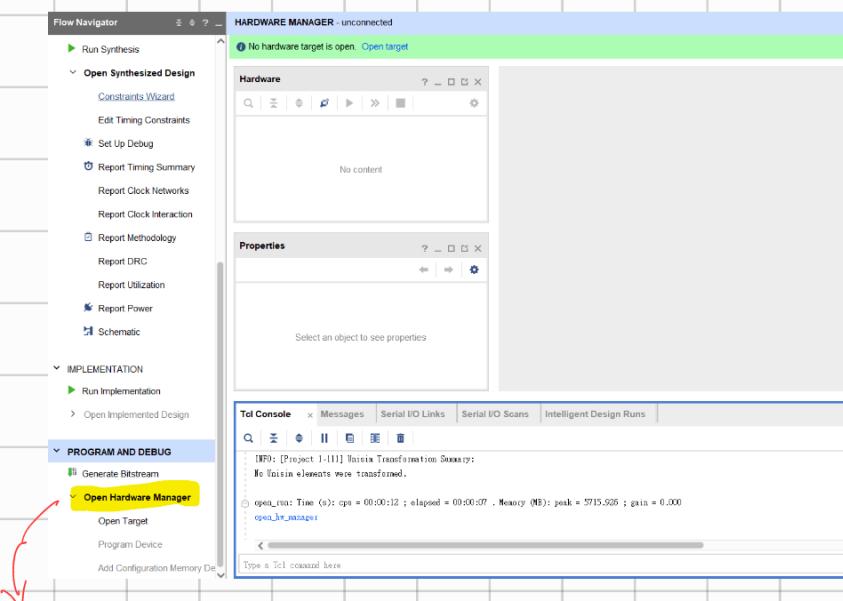


# Vivado logic Analyzer

主要用途為 Verification

Debugging

Data Capture



VLS 主要包涵兩部份

抓 triggering & Data Collection

Debug Corey ILA (integrated logic analysis), System ILA, VIO ... etc

Debug flow HDL instantiation & netlist insertion

## Debug Flow

### HDL Instantiation Flow

The HDL instantiation flow gives the user the **flexibility to connect to any signal** in the design

The user has to modify the HDL code in order to instantiate the cores in the design

### Netlist Insertion Flow

The netlist insertion flow is easier to **add and remove the debug core** and is the more recommended flow

The Vivado tool automatically inserts the debug core into the post-synthesis netlist

## Triggering

Debug 的方式，可以設定要 trigger 的條件，並觀察某印在此條件下的波形表現

在 IP catalog 中提供了 ILA (Integrated logic Analysis) cores，其中都可以做 triggering

多個不同的 ILA，可同時使用

## Setting up Triggering



要在何時抓

Core status: Idle Pre-Trigger Waiting for Trigger Post-Trigger Full

Capture Mode Settings

Capture mode: ALWAYS Number of windows: 1 [1 - 32768] Window data depth: 32768 [1 - 32768] Trigger position in window: 8192 [0 - 32767]

General Settings Refresh rate: 500 ms

Trigger Setup - hw\_ilia\_1 x Capture Setup - hw\_ilia\_1

Name	Operator	Radix	Value	Port
rx_data_rdy	==	[B]	R	probe...

Waveform - hw\_ilia\_1

ILA Status: Idle

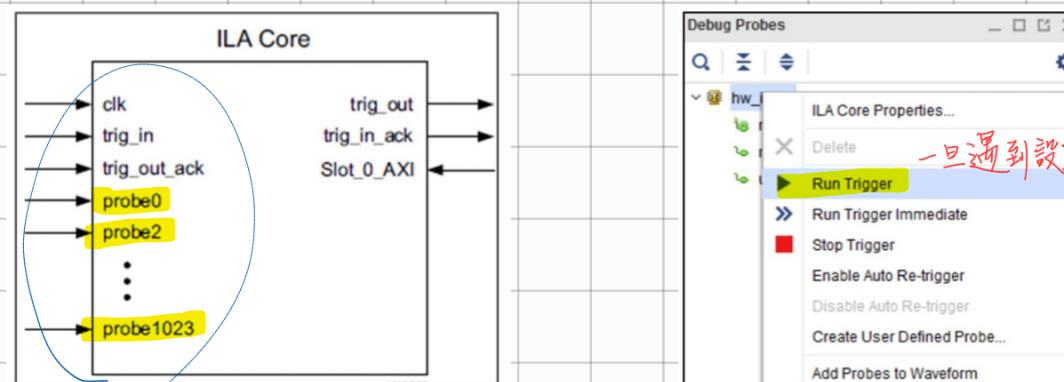
Name	Value
> rx_data[7:0]	41
# uart_rx_i_d[0..n]	1
# rx_data_rdy	0

新增想要 Debug 的 probes  
探針

Trigger Setup - hw\_ilia\_1 Capture Setup - hw\_ilia\_1

Set Capture Condition to Global AND' Set Capture Condition to 'Global OR' Set Capture Condition to Global NAND' Set Capture Condition to 'Global NOR'

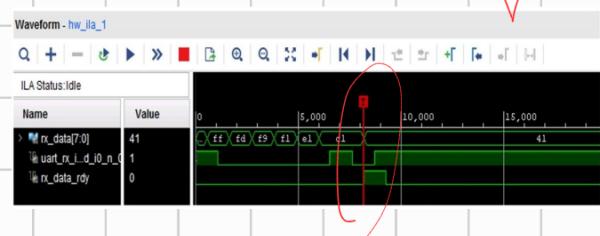
設定 Capture Condition



一個 ILA core 可以設定最多 1024 根探針  
每根 width 最多至 4096

Debug Probes

- Run Trigger
- Run Trigger Immediate
- Stop Trigger
- Enable Auto Re-trigger
- Disable Auto Re-trigger
- Create User Defined Probe...
- Add Probes to Waveform

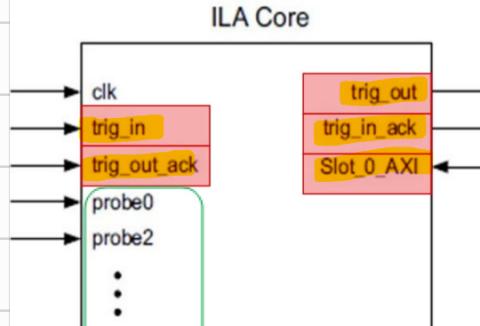


## Integrated Logic Analyzer (ILA)

監測欲 debug 的訊號波形，

目前 monitor type 可設定為 Native 和 AXI 兩個介面

Communication 則是 JTAG



trig\_in 用在加 Zynq 類的 mpsoc, 可連接其它的 ILA 做疊加

trig\_out\_ack 會向其它 ILA core 表示 trig\_out 已正常接收, 此時

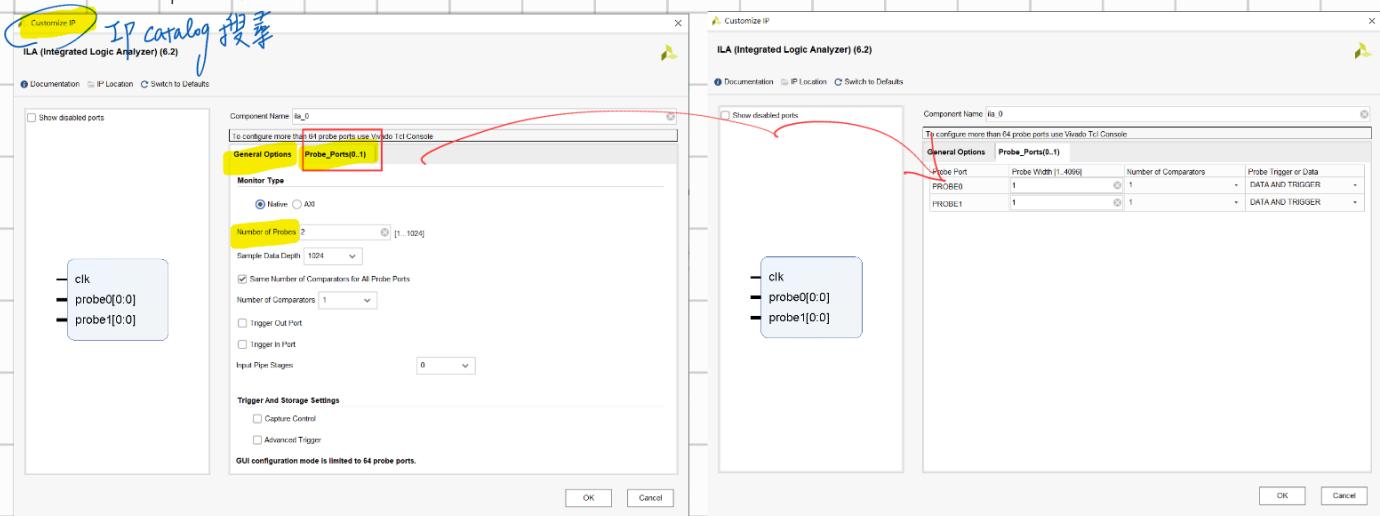


就會降低 trig\_out 訊號

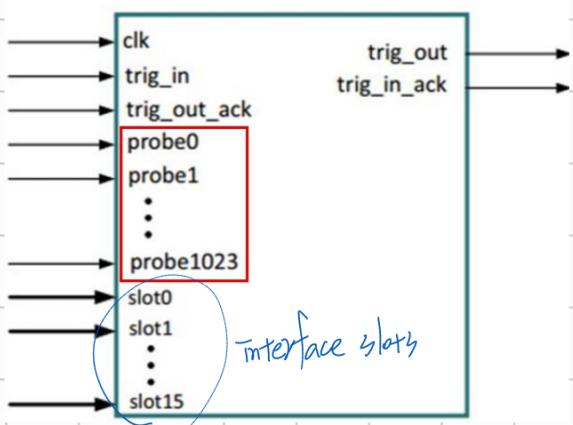
trig\_in\_ack 同樣會向其它 ILA cores 表示 trig\_in 已正常接收

slot\_0\_AXI 會接到 AXI介面

ILA 創建介面

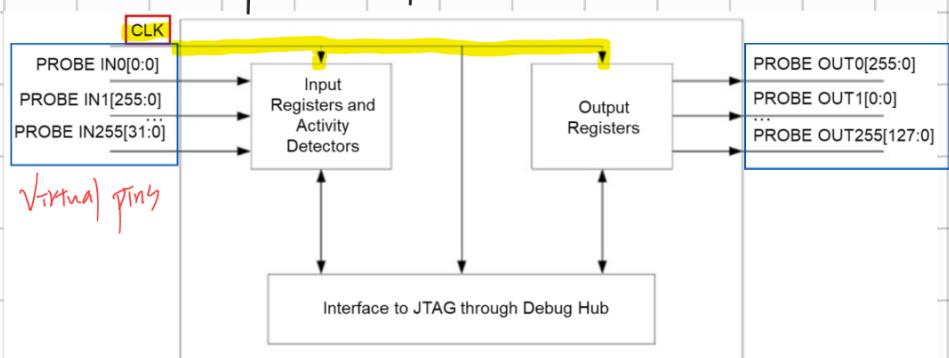


System ILA Core



用來監測內部訊號和介面

Virtual Input / Output (VIO)

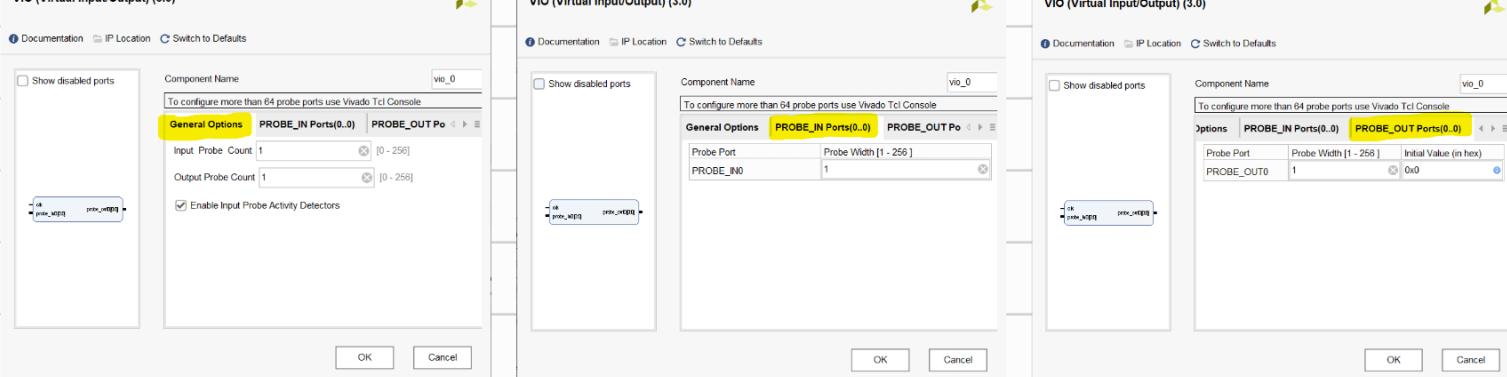


CLK

用於 input/output ports 的 clock

可 Real-Time 監測和驅動 JTAG 內部訊號，不像 ILA 一樣，VIO 不需要 BRAM 去做儲存

VIO 創建介面



只能支援 HDL instantiation flow

## VIO Dashboard

Name	Value	Action	Direction	VIO
rx_data[7:0]	[H] 6C		Input	hw_vio_1
rx_data[7]	●		Input	hw_vio_1
rx_data[6]	●		Input	hw_vio_1
rx_data[5]	●		Input	hw_vio_1
rx_data[4]	●		Input	hw_vio_1
rx_data[3]	●		Input	hw_vio_1
rx_data[2]	●		Input	hw_vio_1
rx_data[1]	●		Input	hw_vio_1
rx_data[0]	●		Input	hw_vio_1
virtual_button	1		Output	hw_vio_1

## Debug Hub

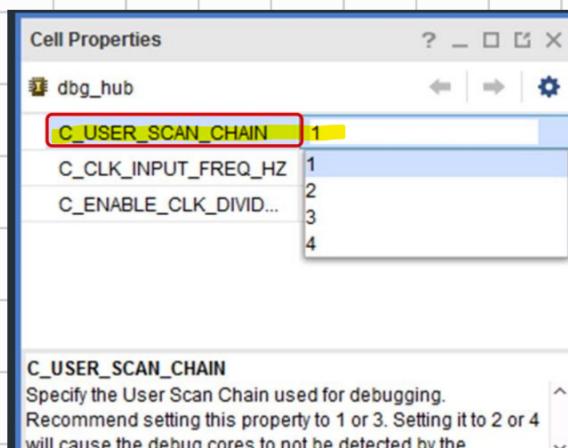
就是 JTAG 和 Debug Cores (ILA, System ILA ... etc) 之間的 Components

當找不到 Debug Core / ILA 時，可能是因為 沒有 clock 或是 clock 沒有運作，JTAG 就無法識別 debug core

② BSCAN\_SWITCH - USER\_MASK 設定錯誤

User Scan chain 用於檢測 debug core，類似通道概念，預設為 1 對應 000

BSCAN 有一端口，BSCAN to JTAG converter 將 BSCAN 作為輸入可從 FPGA 內 / 外部驅動，並提供一 JTAG 做為輸出



# Debug Bridge

為 controller 提供多個選項來與 debug cores 溝通

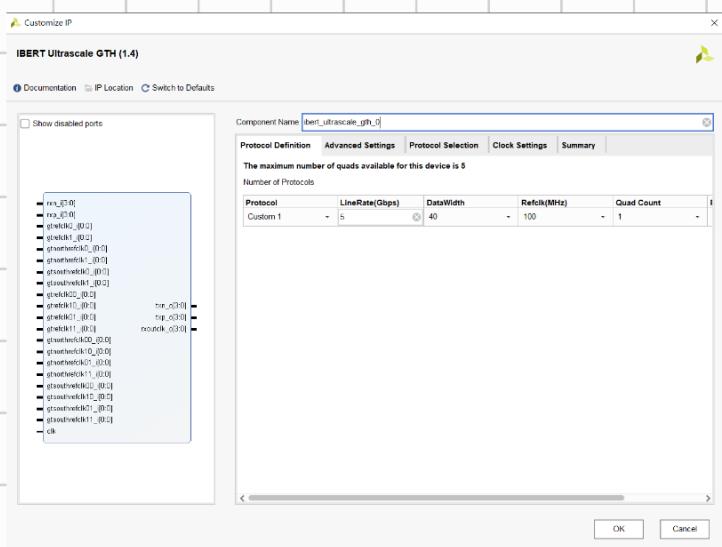
用途 ① 利用 Xilinx Virtual cable (XVC) 透過網路端的方式來遠端 Debug

② for random debugging with field update designs in Dynamic Function Exchange flow ??

# Integrated Bit Error Ratio Tester (IBERT)

能對板子進行高速串行收發器 (transceiver) 的訊號測試

可在 MGT / GTP / GTX / GTH channels 進行 bit-error ratio 的分析



# JTAG to AXI Master

描述了 Master (主設備) 和 Slave (從設備) 之間的數據傳輸方式

此 IP 可作為 AXI Master 來驅動 AXI Slave 外部設備，支持 AXI 和 AXI-Lite 協議

大量數據傳輸

