

~~June 14, 2020~~

10/25/2024

**A simple quick revision was done on 10/25/2024.**

Chipscope\_Intro\_using\_Vivado\_ILA.pdf

ILA = Integrated Logic analyzer

# **Logic Analyzer and On-Chip Logic Analyzer**

## **•Logic Analyzer**

**Digital**

**32/64/128/256 channels**

**Trigger and Storage qualifications**

**vs.**

## **Oscilloscope**

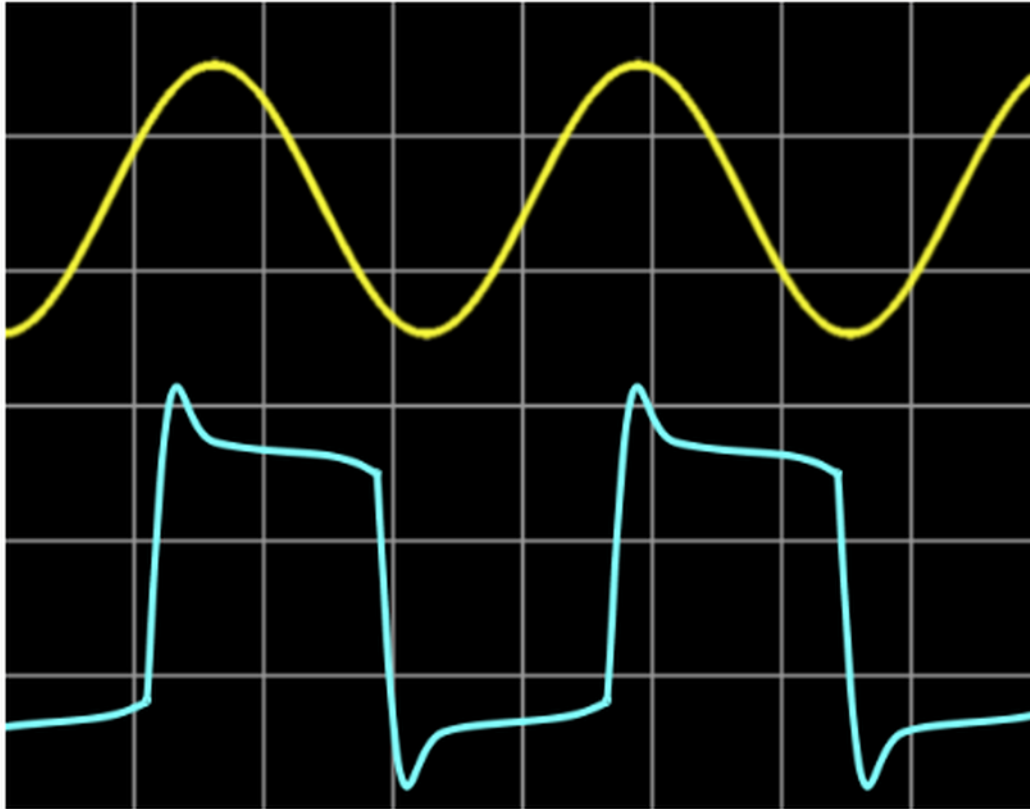
**Analog**

**2 or 4 channels**

**Simple trigger**

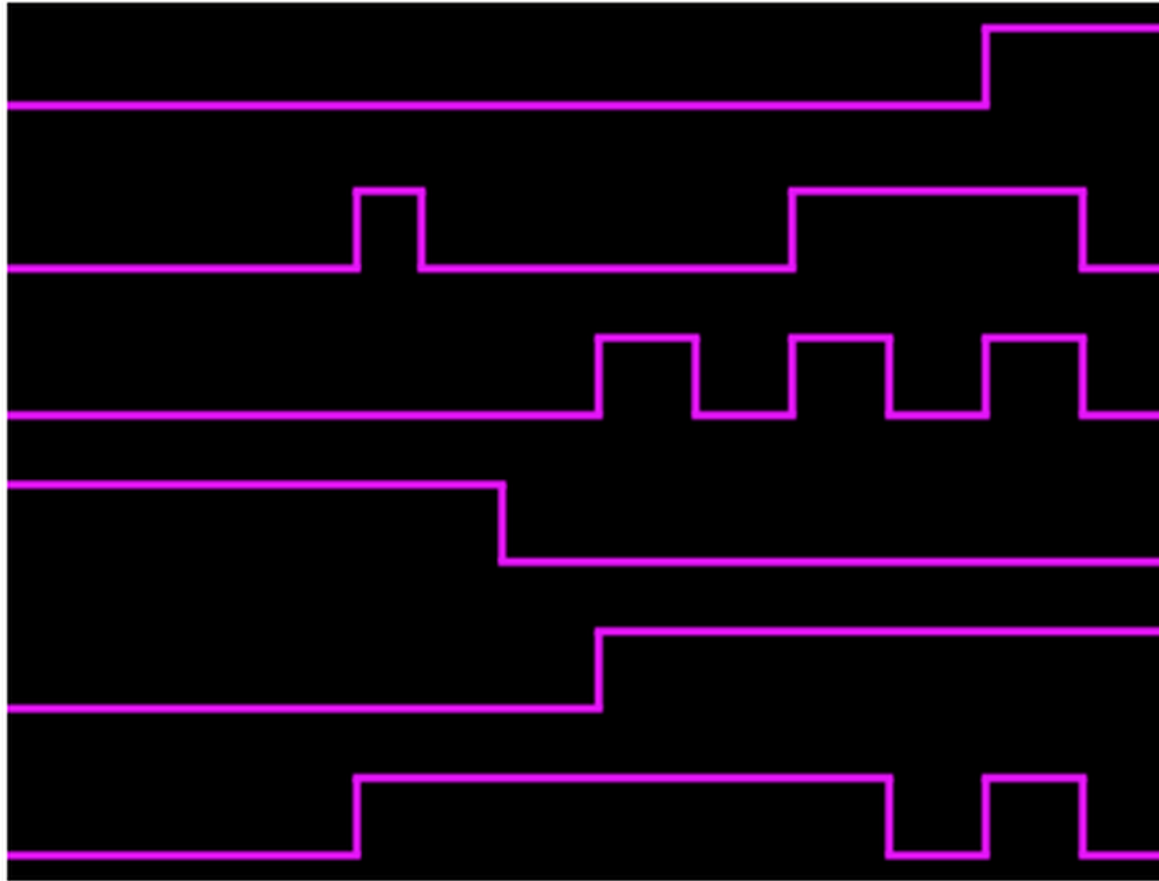
- SoC (System on Chip) and Embedded Processors in SoC**

<https://articles.saleae.com/logic-analyzers/logic-analyzer-vs.-oscilloscope>



Example of an oscilloscope displaying a sine wave and overshoot on two channels

<https://articles.saleae.com/logic-analyzers/logic-analyzer-vs.-oscilloscope>



Example of a logic analyzer displaying digital signals on six channels



# References:

<https://docs.amd.com/r/en-US/ug908-vivado-programming-debugging/ILA>

<https://docs.amd.com/r/en-US/ug936-vivado-tutorial-programming-debugging/Target-Board-and-Server-Set-Up>

## EE354L Detour lab

[https://ece-classes.usc.edu/ee254/ee254l\\_lab\\_manual/detour\\_signal\\_schematic/ee354l\\_detour\\_rev7.pdf](https://ece-classes.usc.edu/ee254/ee254l_lab_manual/detour_signal_schematic/ee354l_detour_rev7.pdf)

# EE201L Detour lab (original)

	GL	G1	G2	GR
Idle State	<○ ○	○ ○	○ ○	○ ○>
R1 State (G1 is ON)	<○ ○	● ●	○ ○	○ ○>
R12 State (G1, G2 are ON)	<○ ○	● ●	● ●	○ ○>
R123 State (G1, G2, GR are ON)	<○ ○	● ●	● ●	● ●>
Idle State	<○ ○	○ ○	○ ○	○ ○>
L1 State (G2 is ON)	<○ ○	○ ○	● ●	○ ○>
L12 State (G2, G1 are ON)	<○ ○	● ●	● ●	○ ○>
L123 State (G2, G1, GL are ON)	● ●	● ●	● ●	○ ○>

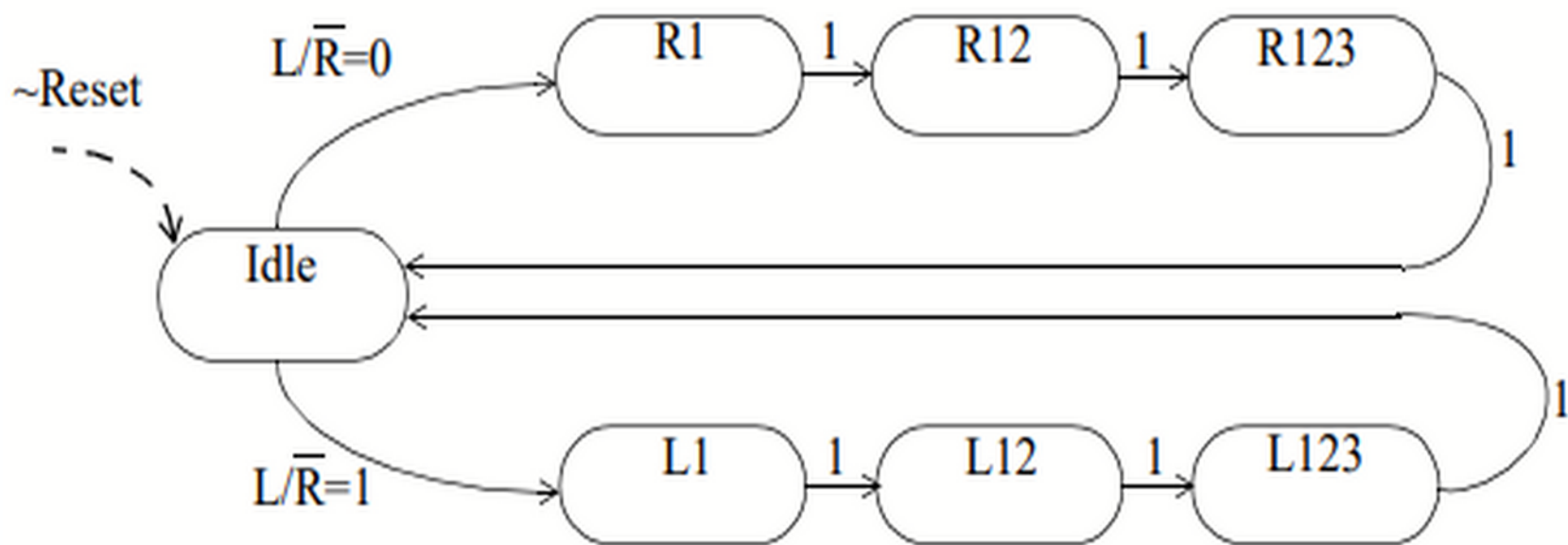


Fig 2: State diagram for the Detour Signal design



# EE201L Detour lab (modified)

		GL	G1	G2	GR	
0	IS0 (Idle Start 0) State	<○	○	○	○	○>
1	IS1 (Idle Start 1) State	<●	●	●	●	●>
2	I (Idle) State	<○	○	○	○	○>
3	R1 State (G1 is ON)	<○	○	●	●	○>
4	R12 State (G1, G2 are ON)	<○	○	●	●	○>
5	R123 State (G1, G2, GR are ON)	<○	○	●	●	●>
6	R123_DIS State (all OFF)	<○	○	○	○	○>
7	R123_REA State (G1, G2, GR are ON)	<○	○	●	●	●>
8	I (Idle) State	<○	○	○	○	○>
9	L1 State (G2 is ON)	<○	○	○	○	○>
A	L12 State (G2, G1 are ON)	<○	○	●	●	○>
B	L123 State (G2, G1, GL are ON)	<●	●	●	●	○>
C	L123_DIS State (all OFF)	<○	○	○	○	○>
	L123_REA State (G2, G1, GL are ON)	<●	●	●	●	○>

Starting LAMP TEST by  
~~16~~ times flashin all lights

8

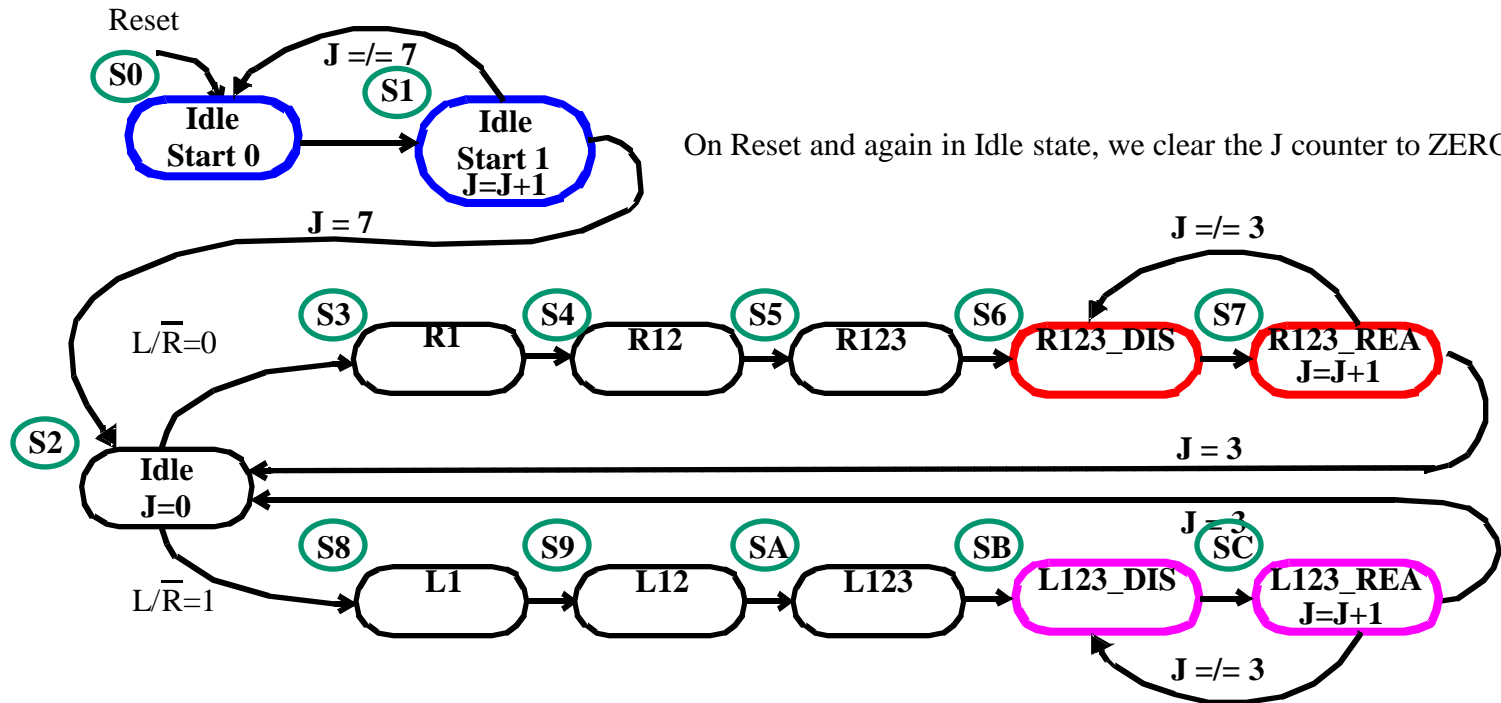
Disappear/Reappear  
sequence ~~16~~ times

4

Disappear/Reappear  
sequence ~~16~~ times

4

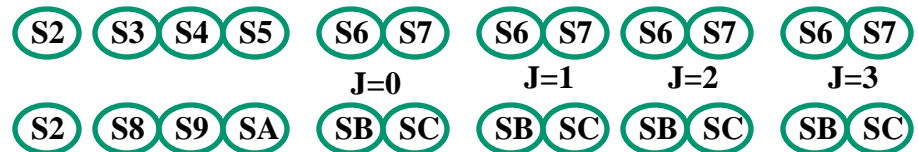
# Modified Detour lab state diagram

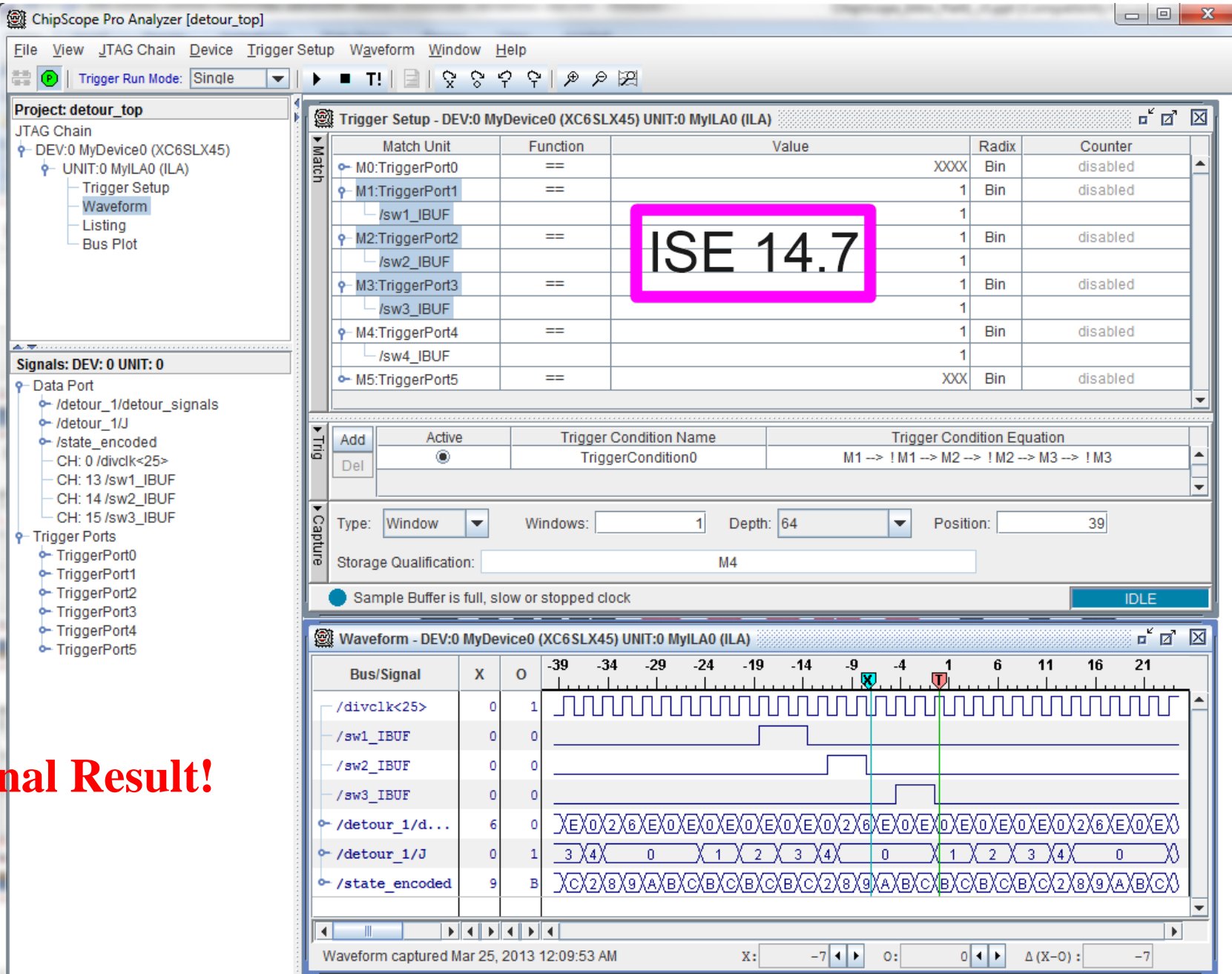


During Reset it remains in S0 state.

After Reset, it does S0 S1 eight times.

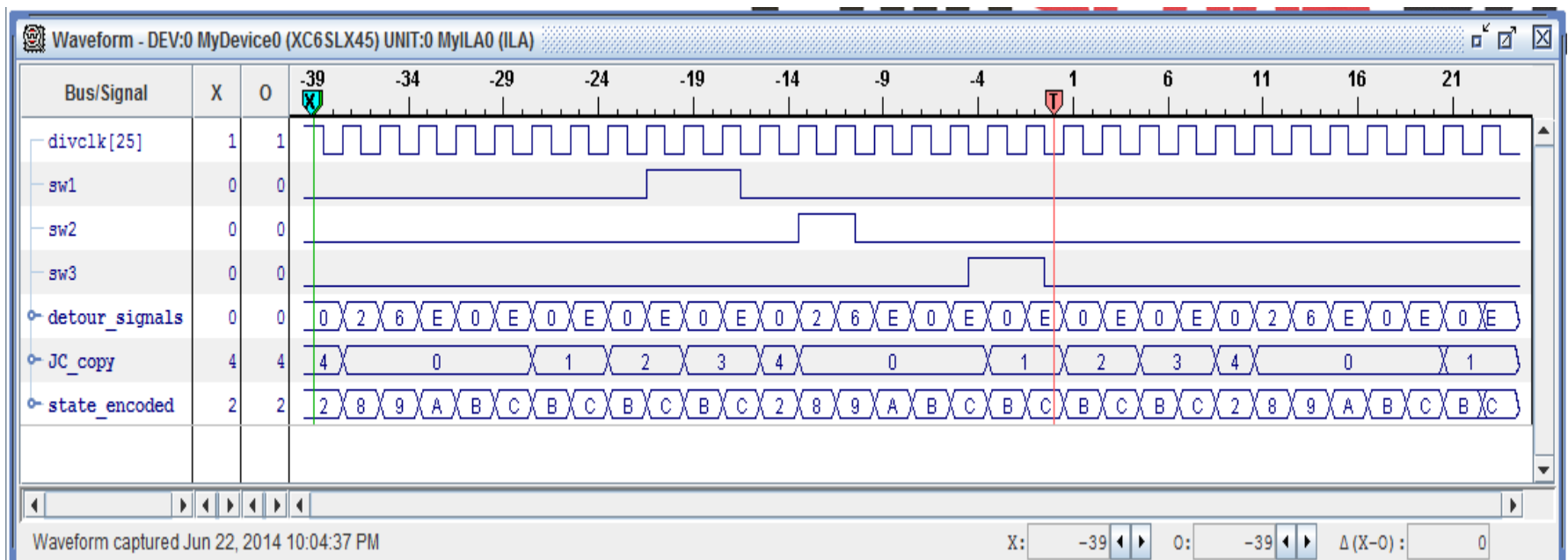
Then it does one of the two depending upon  $\overline{L/R}$ .





Final Result!

# ISE 14.7



# Some advanced topics

- Sampling on Rising or Falling edge
- Different Match Units
- Complex Trigger Conditions
- Virtual I/O

# Can you see the clock?

- You can NOT see the **sampling clock**!
- To see **system clock** on the waveform, you need to use a trigger clock (sampling clock) with **higher frequency** (higher than the system clock, at least **double**).
- Most designers do not need to see the clock!

# Sample on Rising or Falling edge?

- Ideally we should sample once per clock at the significant edge of the clock.
- Same edge sampling → The Sampling process should not introduce delay. You should capture data at *or just before the significant edge*, **not** just after the significant edge. *If it is just after the edge, you will be capturing data in transition (data in turmoil) !*

# Sample on Rising or Falling edge?

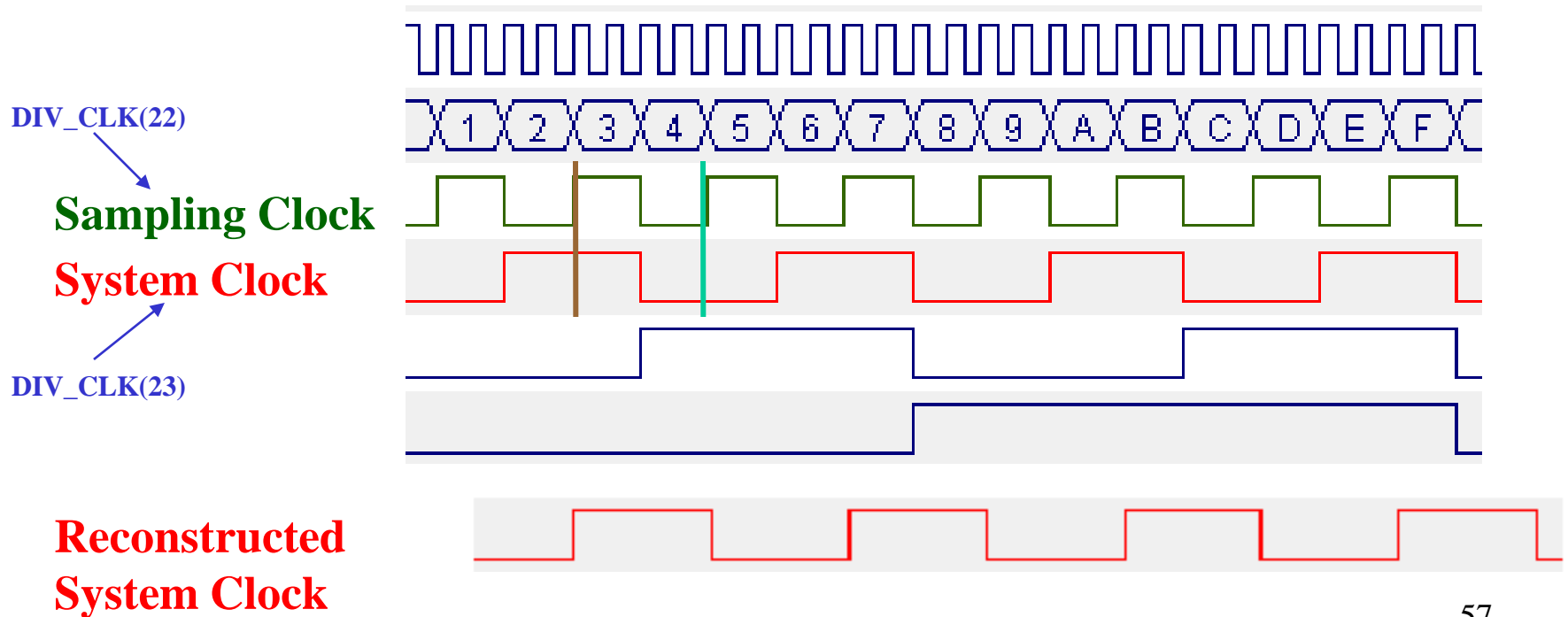
in this detour demo for EE354L and EE560

- Here/we are sampling using high-speed clock (higher speed than the system clock).  
Actually double the clock rate.
- Here we are running at a low speed
- So it *safer* to sample somewhere in the middle of the clock.



# Sample on Rising or Falling edge?

Two middles (**one-fourth** and **three-fourth** points) of the system clock coincide with the RISING edge of the high-speed clock.



# Procedure to add an ILA IP to your project:

1. It is better to complete your design first without a chipscope (without and ILA debug probe)

ISE 14.7

2. Then add an ILA core from the IP Catalog.  
Search for ILA.

The screenshot illustrates the steps to add an ILA IP in Xilinx ISE 14.7. It shows the 'Flow Navigator' on the left, the 'IP Catalog' window in the center, and the 'Debug & Verification' pane at the bottom.

**Step 1:** In the 'Flow Navigator', the 'IP Catalog' option under 'PROJECT MANAGER' is highlighted with a red arrow and a circled '1'.

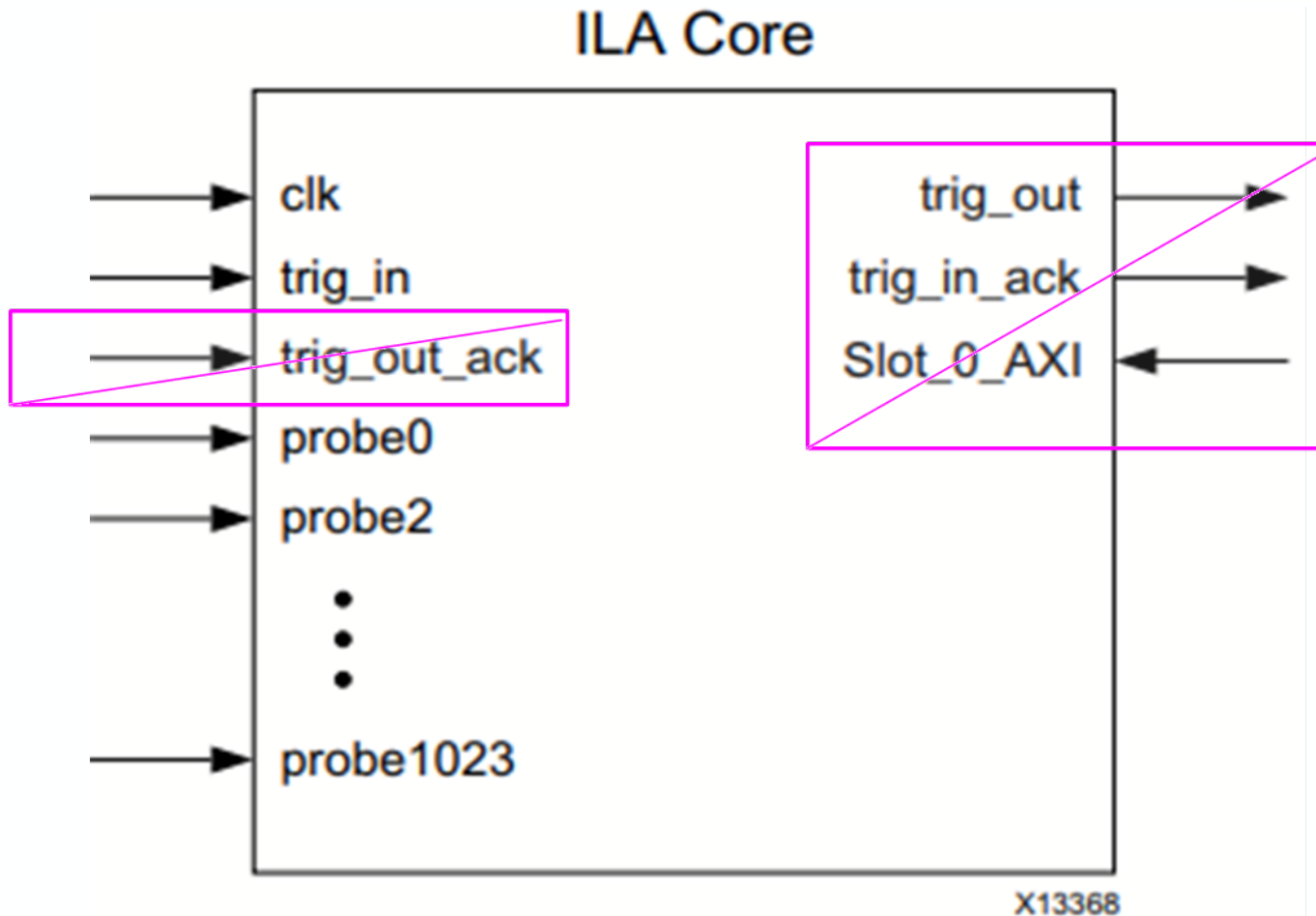
**Step 2:** In the 'IP Catalog' window, the search bar contains 'ILA', highlighted with a red arrow and a circled '2'. The search results show '(4 matches)'.

**Step 3:** In the 'Debug & Verification' pane, the 'ILA (Integrated Logic Analyzer)' entry is highlighted with a red arrow and a circled '3'.

Component	Version	Category	Status	Source
ILA (Integrated Logic Analyzer)	AXI4, AXI4-Stream	Production	Included	xilinx.com:ip:ila:6.2

### 3. Configure the ILA to suit your needs.

ISE 14.7



*Figure 1-1: ILA Core Symbol*

