Welcome to The Hardware Design & Lab!

Fall 2024
Keyboard Tutorial

Prof. Chun-Yi Lee

Department of Computer Science
National Tsing Hua University

Agenda

- Announcement
- Keyboard Tutorial
  - Basic protocols
  - KeyboardCtrl.v
  - KeyboardDecoder.v

Today's class will help you:

1. Understand the protocol of keyboards
2. Understand how to use the provided keyboard control codes
3. Understand how to use the keyboard IP

Announcements

- Lab 5
  - Lab 5 basic questions due on **10/31/2024 (Thu)**
  - Lab 5 advanced questions & FPGA demonstration on **11/7/2024 (Thu)**
  - **Please read the lab description and specification carefully**
  - **Please follow the template I/Os and submit your .v files**
- Final project
  - You can begin working on your final projects from now
  - Please avoid copying the works from past years

Final Project

- Use your FPGA to implement creative and interesting works

- Final project proposal due on **11/25/2024 (Mon)**
- Final project report due on **12/19/2024 (Fri)**

- Final exam
  - **12/12/2024 (Thu)**
  - 2 hours

Final Project Presentation

- Final project demonstration on **12/17/2024 (Tue)** and **12/19/2024 (Thu)**
  - Around 10 minutes per team

- **No restriction on your presentation style**
  - Be creative!
  - What is special and new in your project
  - Key features

- Order of presentation
  - We will randomly decide the order
  - Let us know if you have constraints

Final Project Report

- Final project report
  - Due on **12/19/2024, 23:59pm (Thu)**
  - Block diagrams and state transition diagrams
  - **Detailed explanation**

- Report contents
  - Introduction
  - Motivation
  - System specification
  - Experimental results
  - Conclusion
  - …And any other sections that you would like to include

Final Project Award

- **Category:** Difficulty and completeness
  - Graded by me and the TAs
    - Difficulty (**35%**)
    - Completeness (**30%**)
    - Source code coding style (**15%**) (Correctness, usage, comments, etc.)
    - Report (**20%**)
  - First place: **NTD $2,000**
  - Second place: **NTD $1,500**
  - Third place: **NTD $1,200**

- **Category:** Best creativity
  - Voted by everyone in the class
  - **NTD $1,000**

- **Category:** Special challenge question using FPGA
  - Extra **bonus points**. **Cash rewards $12,000 for the best one**.

Final Exam

- Dates and time
  - **12/12/2024 (Thu)**
  - **3:30pm ~ 5:30pm (2 hours)**

- Course contents
  - Verilog design questions
  - Logic design concepts
  - Lecture & lab contents

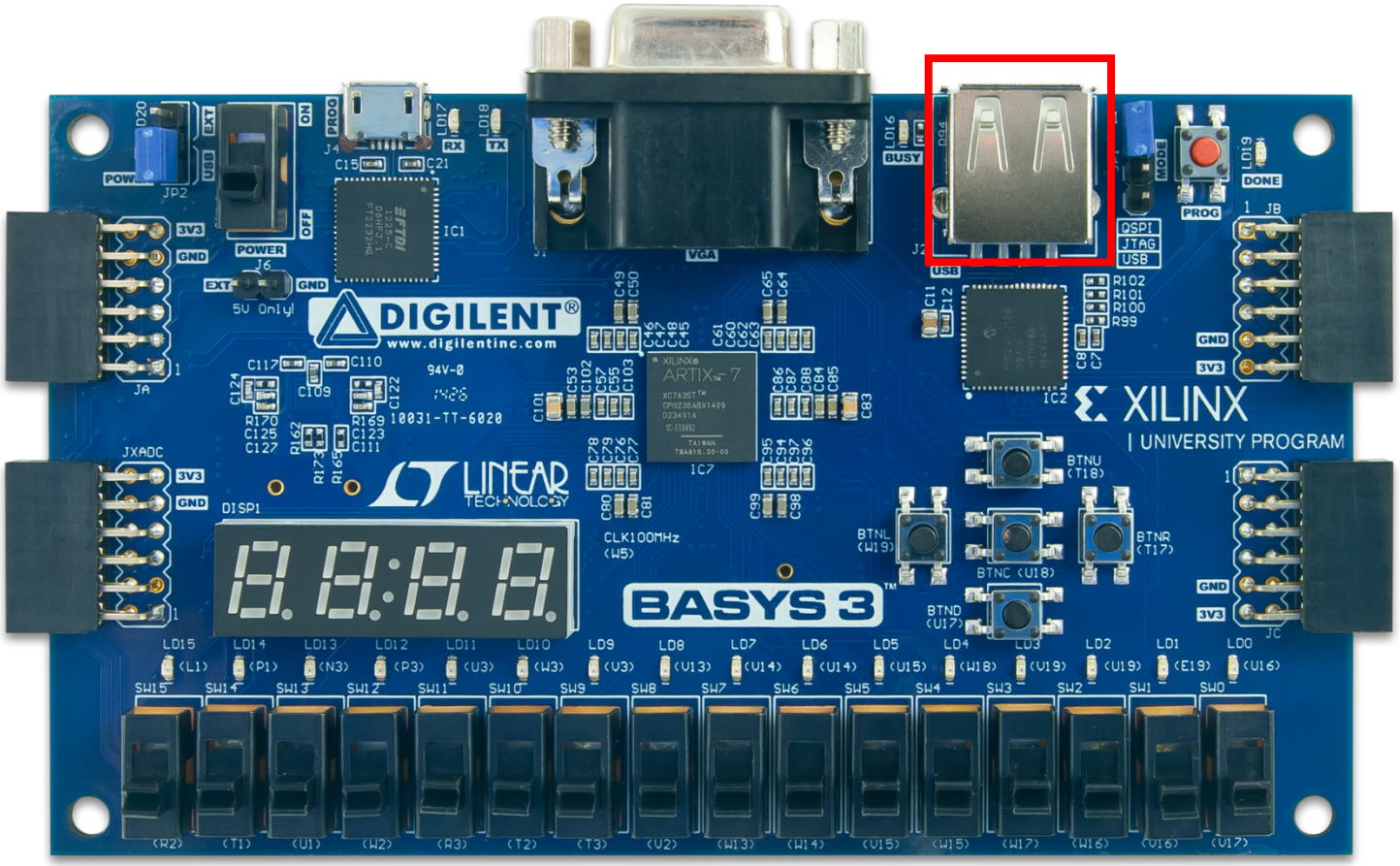
Agenda

- Announcement
- Keyboard Tutorial
  - Basic protocols
  - KeyboardCtrl.v
  - KeyboardDecoder.v

Today's class will help you:

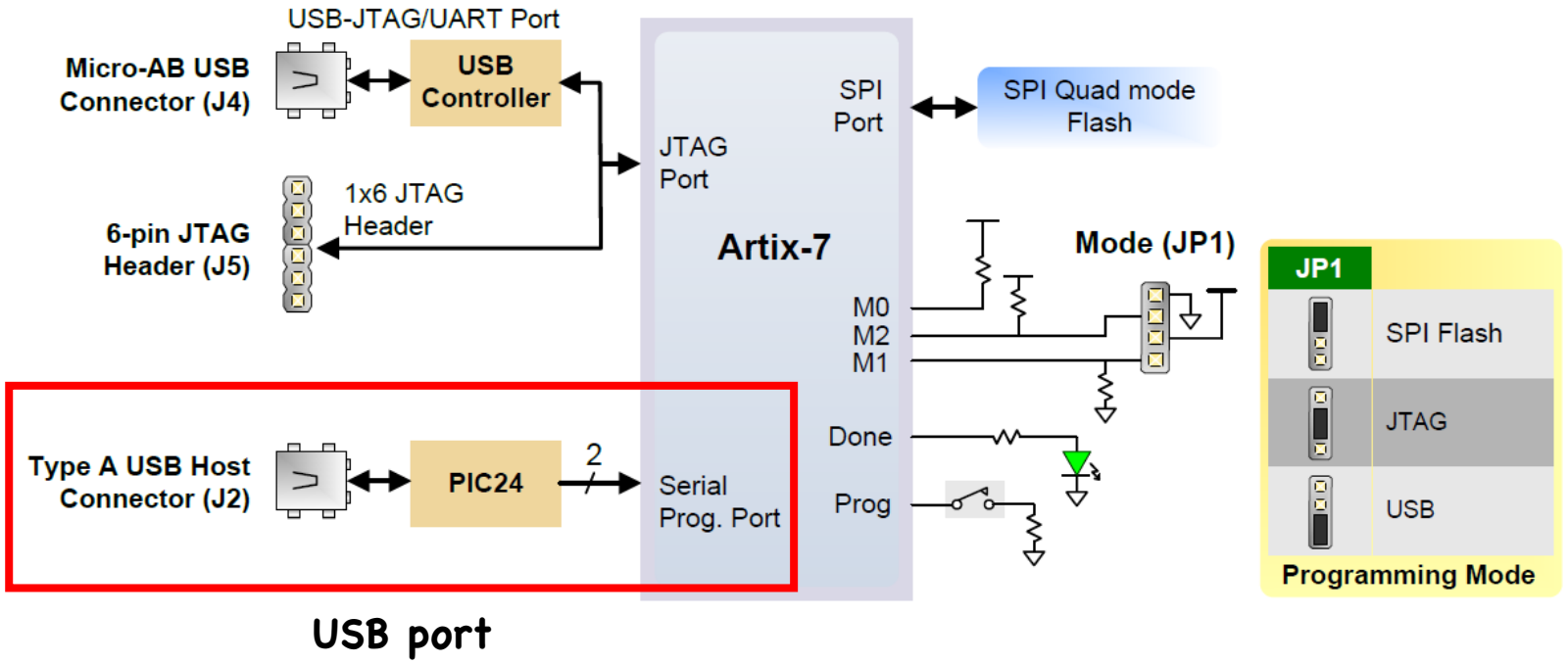
1. Understand the protocol of keyboards
2. Understand how to use the provided keyboard control codes
3. Understand how to use the keyboard IP

USB HID Host (1/3)



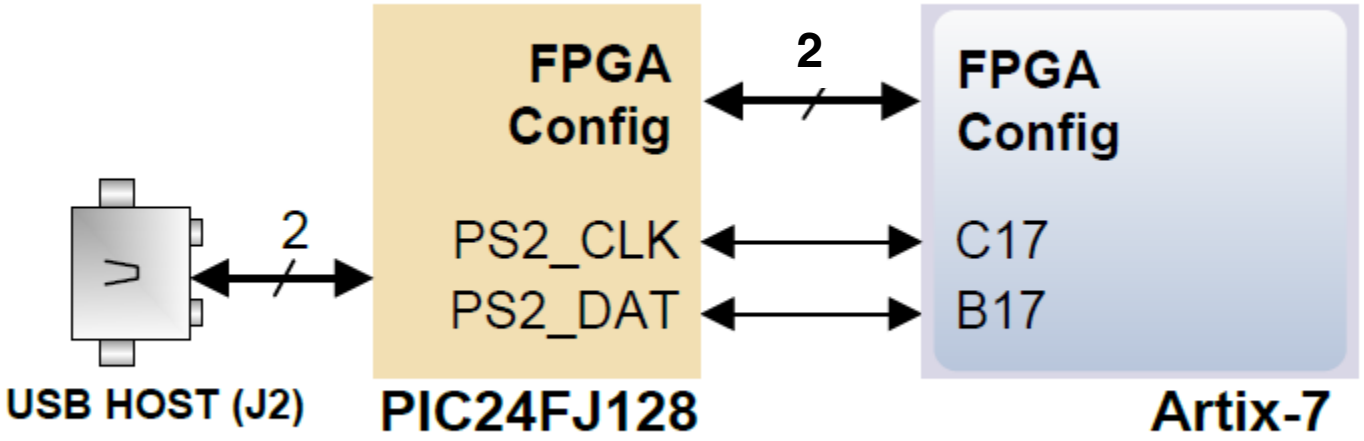
USB HID Host (2/3)

- Two bit serial port is used by Basys 3 to connect USB



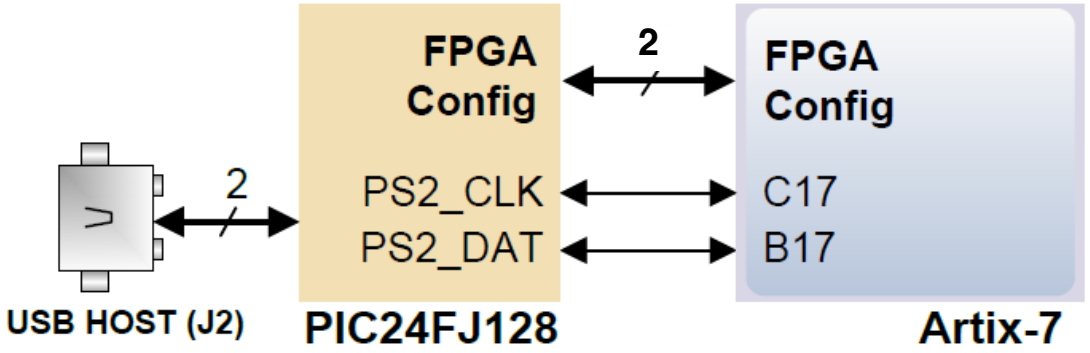
USB HID Host (3/3)

- The USB port is controlled via a PIC24 chip
  - The PIC24 chip translates USB signals to PS2 signals
  - The two interface ports on the FPGA are **B17** and **C17**



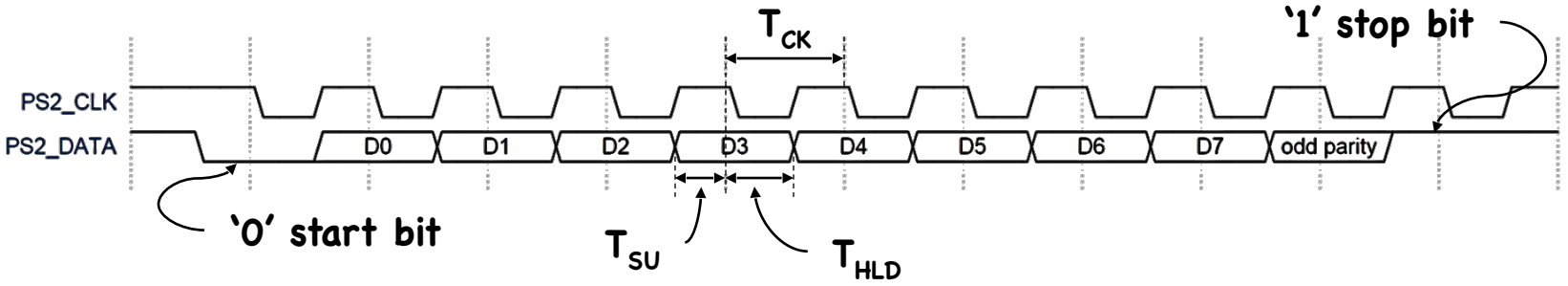
Microchip PIC24FJ128

- Configuration mode
  - Download a bitstream to the FPGA
- Application mode
  - On Basys 3, it is called USB HID Host mode
  - **Hub support is not currently available**
  - Only a single mouse or a single keyboard can be connected
  - PS2_CLK and PS2_DATA are used to implement **a standard PS/2 interface**



HID Controller

- Every time the keyboard transmits 8 bits to FPGA
  - The 8 bits are followed by a parity bit
  - Each bit is read on the falling edge of the clock



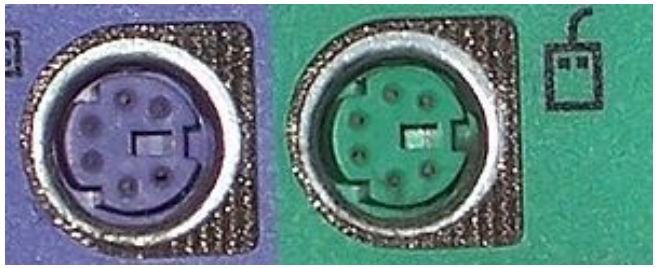
Symbol	Parameter	Min	Max
T_{CK}	Clock time	30 us	50 us
T_{SU}	Data-to-clock setup time	5 us	25 us
T_{HLD}	Clock-to-data hold time	5 us	25 us

Initialization

- When a keyboard or mouse is connected to the Basys 3, a “self-test passed” command (**0xAA**) is sent to the Basys 3.
  - This command (**0xAA**) is automated sent by the keyboard to FPGA
  - The command informs FPGA that the keyboard is ready
  - Only sent once when the system turns on
- The Basys 3 board may request the keyboard to do several times of self-test by command (**0xFF**)
  - The keyboard uses **0xFA** to respond to the requests, and may send “self-test passed” again to Basys 3
  - Keyboard : 0xFA (0xFA → 0xAA)
- The protocol is already implemented and will be provided to you
  - However, it is still good to know about it

PS/2 Port

- PS/2 port was formerly used for keyboards and mouses
- Nowadays, they are replaced by USB ports
- Their protocols are still widely used
- Scancodes are used to communicate with the PS/2 port



Example PC compatible (IBM PS/2) scancodes

key	set 1 (IBM PC XT)		set 2 (IBM PC AT)		set 3 (IBM 3270 PC)	
	press	release	press	release	press	release
A (normal letter)	1E	9E	1C	F0 1C	1C	F0 1C
Return / Enter (main keyboard)	1C	9C	5A	F0 5A	5A	F0 5A
Enter (numeric keypad)	E0 1C	E0 9C	E0 5A	E0 F0 5A	79	F0 79
Left Windows key	E0 5B	E0 DB	E0 1F	E0 F0 1F	8B	F0 8B
Right Windows key	E0 5C	E0 DC	E0 27	E0 F0 27	8C	F0 8C

from Wiki

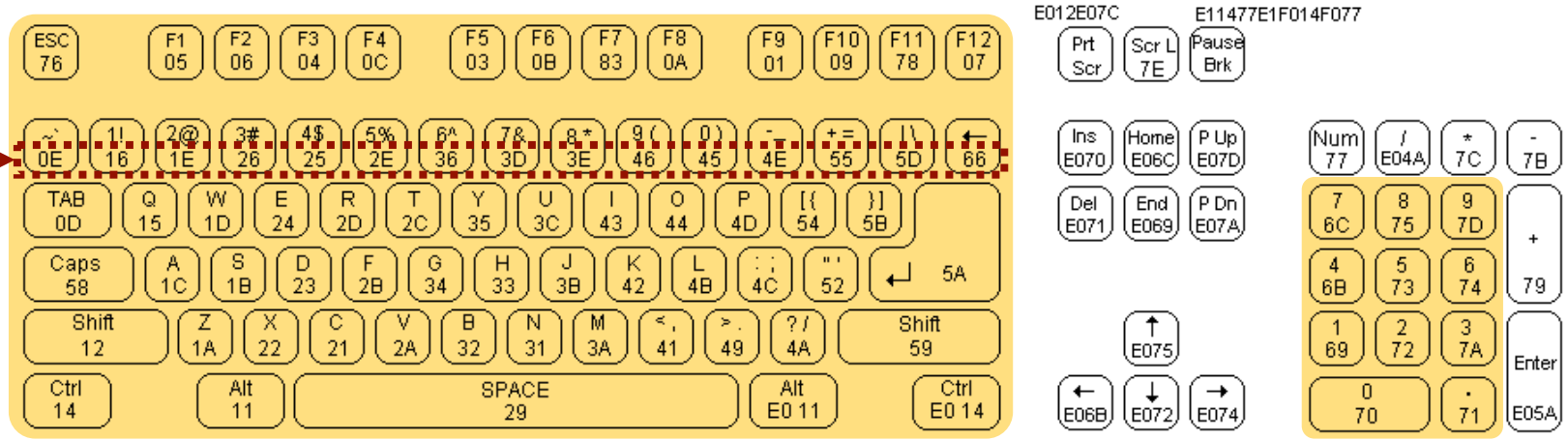
PS/2 Scancode

- Three types of scancodes
  - Make codes represent the key values
  - Break code represents the action of "release the key"
  - Extend code represent the **duplicate** of a key

Extend Code	Break Code	Make code
E0	F0	XX

(means "release")

Make code

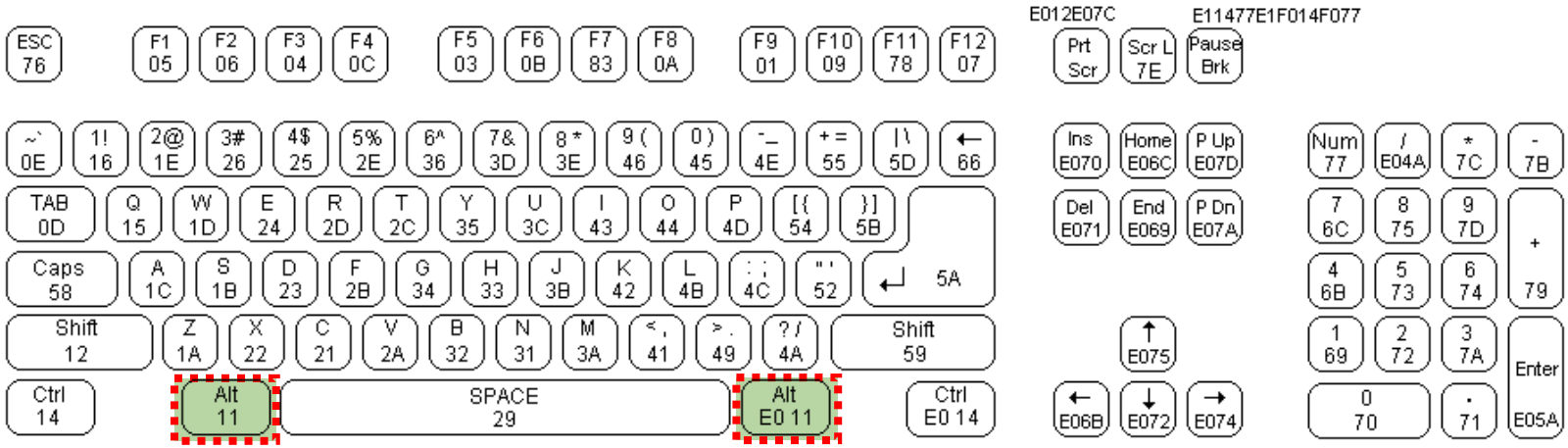


We only use the yellow parts of the keyboard

PS/2 Scancode (Example)

- Two "Alt" key are available on the keyboard
  - Additional keys are encoded as <**extend code**> + <**make code**>
- When releasing a key, a <**break code**> is inserted

L Alt press			11
L Alt release		F0	11
R Alt press	E0		11
R Alt release	E0	F0	11



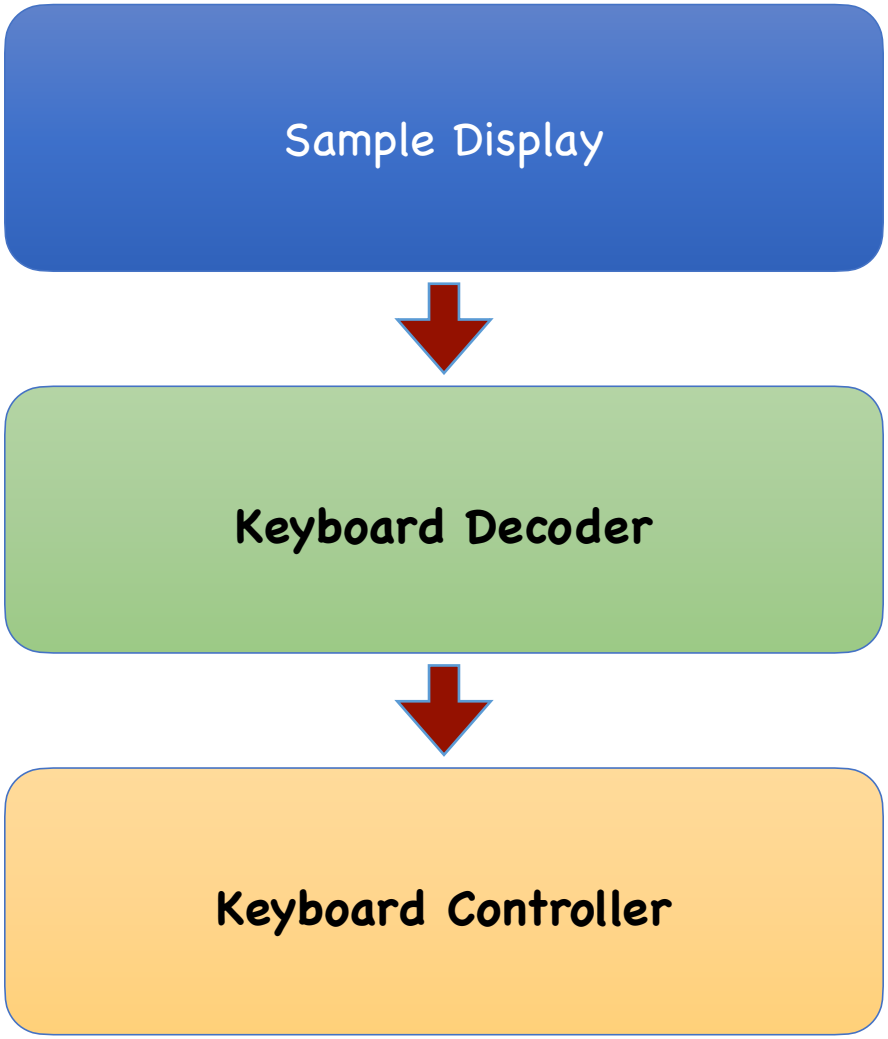
Agenda

- Announcement
- **Keyboard Tutorial**
  - Basic protocols
  - **KeyboardCtrl.v**
  - **KeyboardDecoder.v**

Today's class will help you:

1. Understand the protocol of keyboards
2. Understand how to use the provided keyboard control codes
3. Understand how to use the keyboard IP

Program Hierarchy



Verilog Module: KeyboardCtrl

- In Keyboard-Controller
  - Ps2Interface.v
  - KeyboardCtrl.v

- I/Os for KeyboardCtrl.v

Output

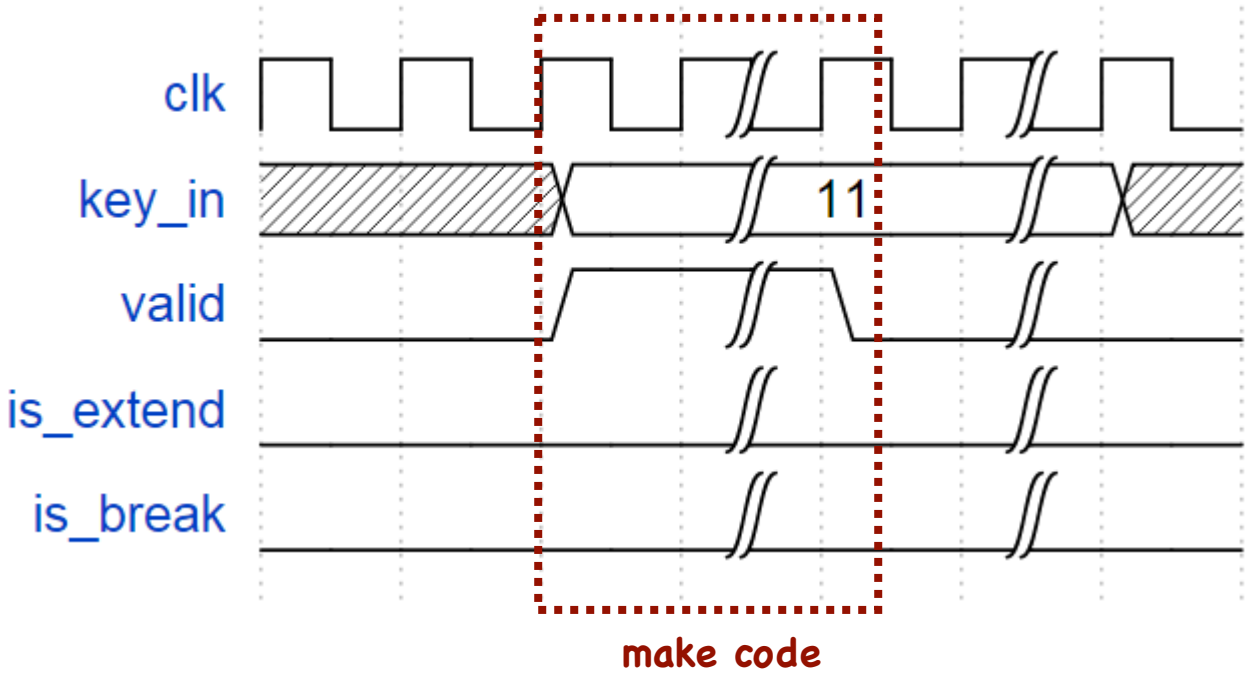
- key_in
- is_extend
- is_break
- valid
- err

Input

- PS2_CLK
- PS2_DATA
- rst
- clk

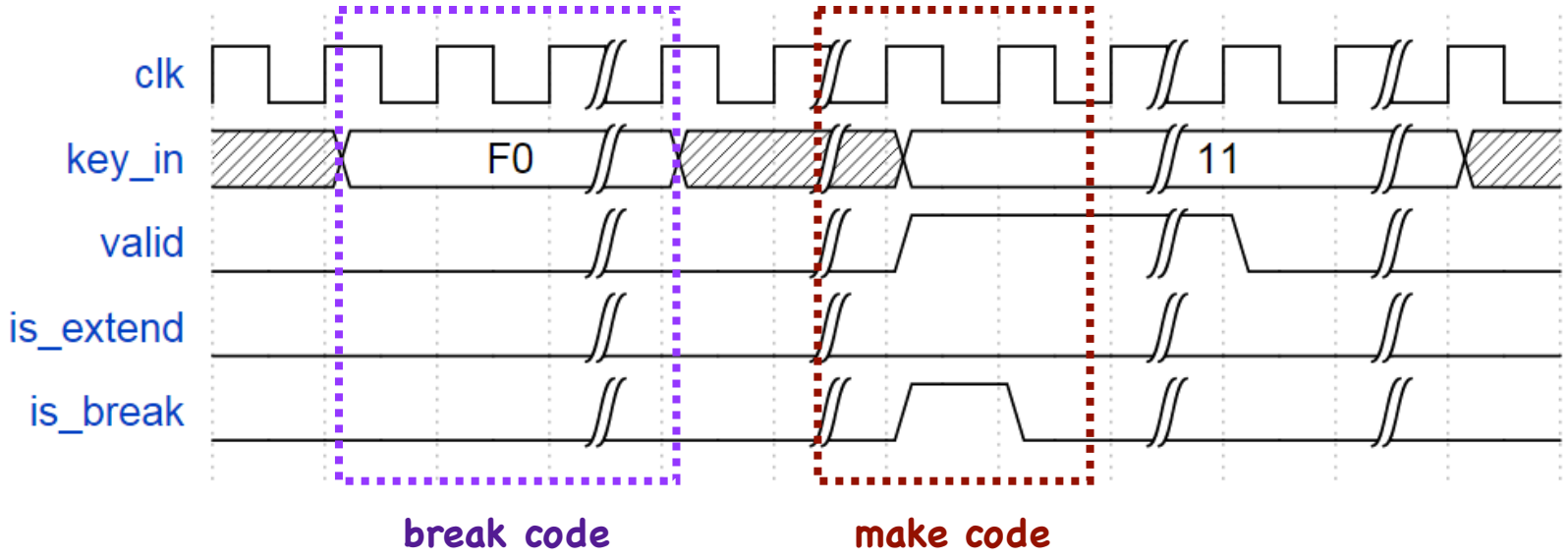
KeyboardCtrl (Output Example 1)

- L Alt press



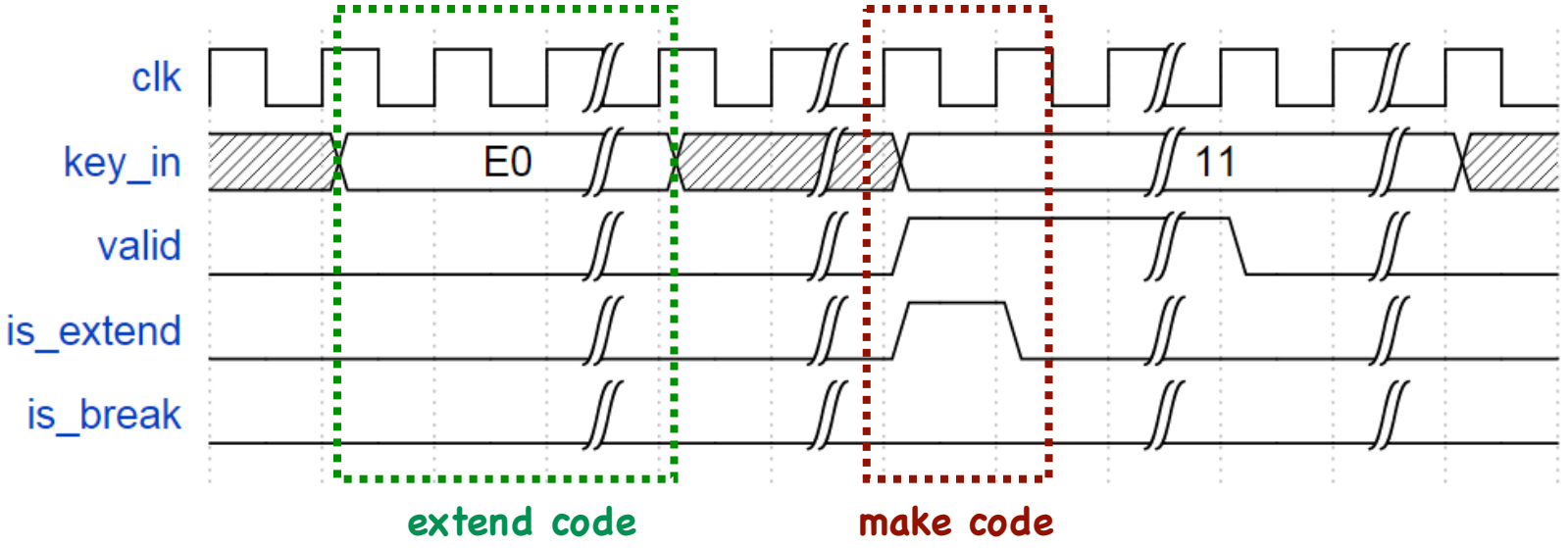
KeyboardCtrl (Output Example 2)

- L Alt release



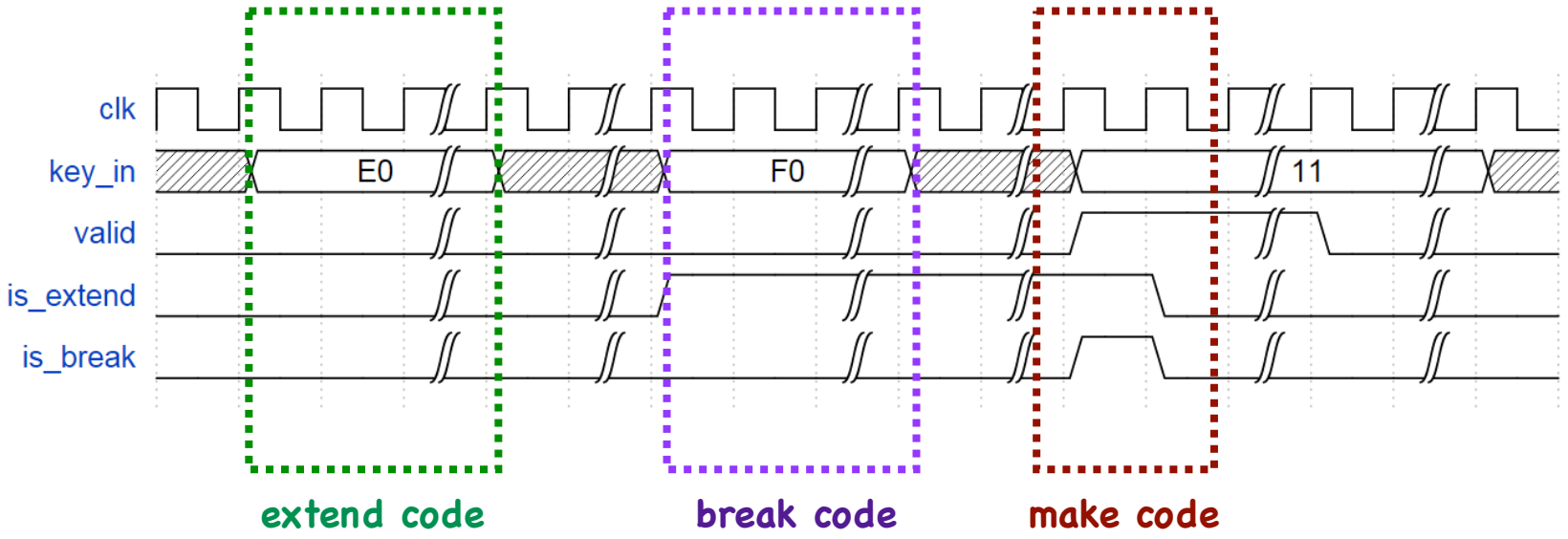
KeyboardCtrl (Output Example 3)

- R Alt press



KeyboardCtrl (Output Example 4)

- R Alt release



Agenda

- Announcement
- **Keyboard Tutorial**
  - Basic protocols
  - KeyboardCtrl.v
  - **KeyboardDecoder.v**

Today's class will help you:

1. Understand the protocol of keyboards
2. Understand how to use the provided keyboard control codes
3. Understand how to use the keyboard IP

Verilog Module: KeyboardDecoder (1/5)

- In Keyboard Sample Code
  - KeyboardDecoder.v
  - SampleDisplay.v

- I/O for KeyboardDecoder

Output

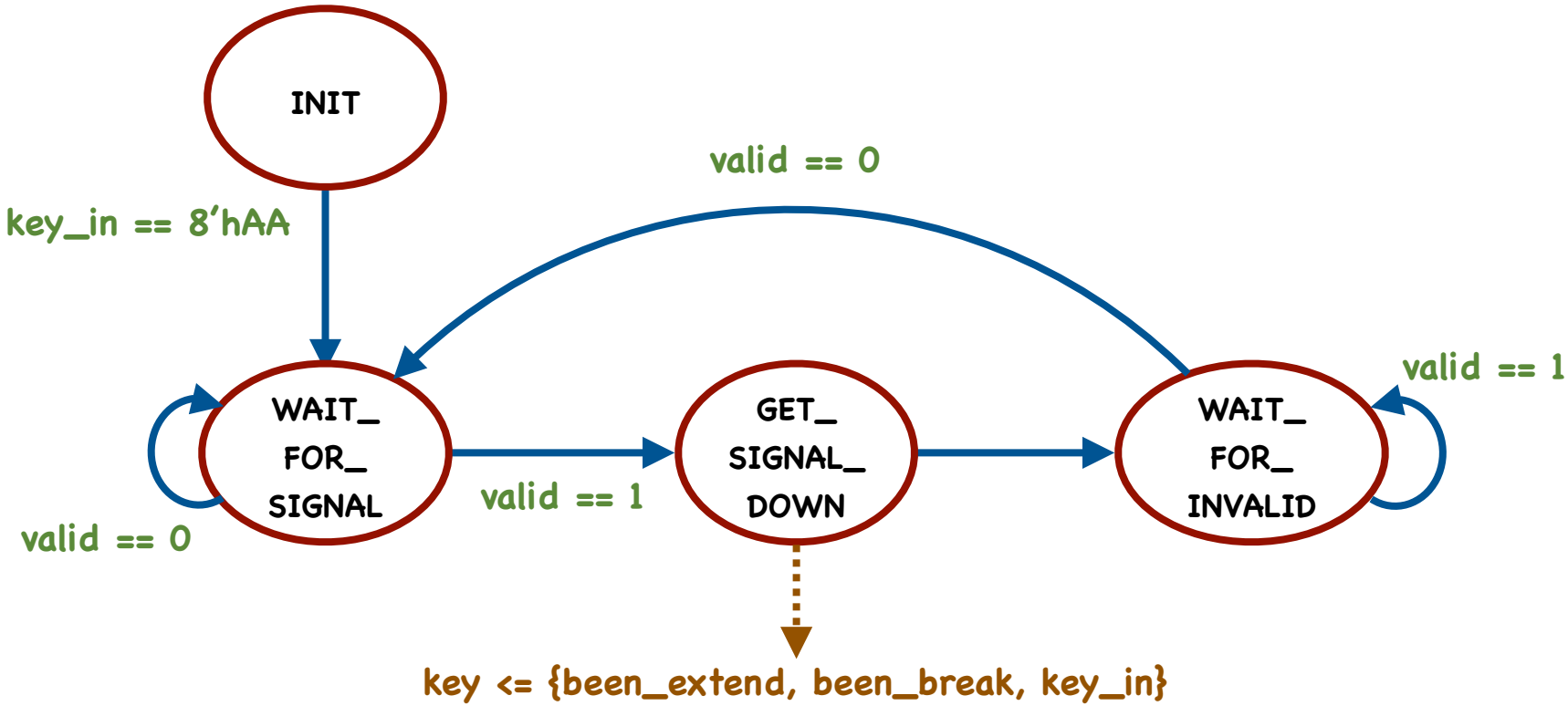
- key_down
- last_change
- Key_valid

Input

- PS2_CLK
- PS2_DATA
- rst
- clk

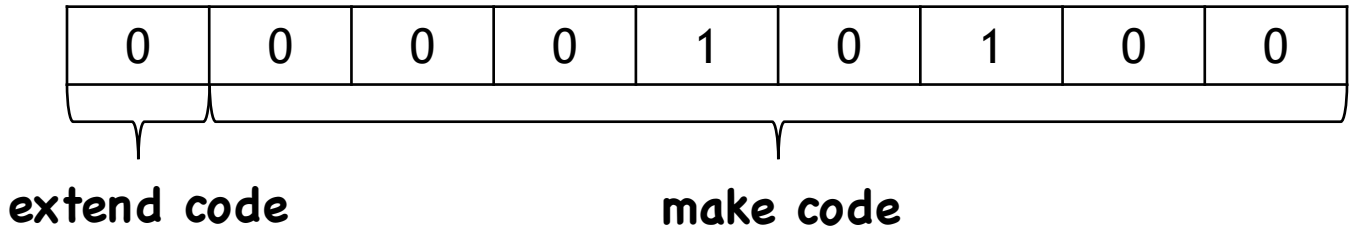
Verilog Module: KeyboardDecoder (2/5)

- Retrieves "**extend**", "**break**", and "**key**" from KeyboardCtrl
  - Gets value only when **valid = 1'b1**



Verilog Module: KeyboardDecoder (3/5)

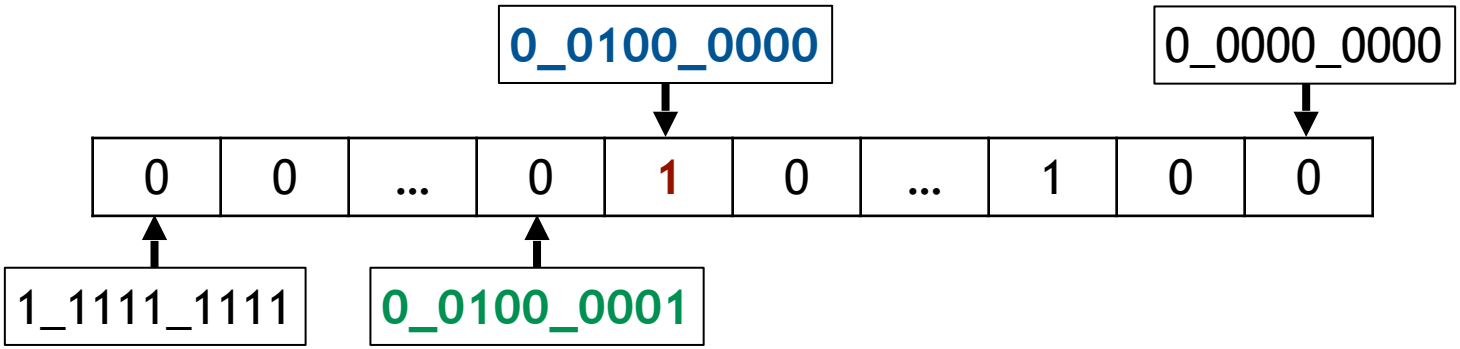
- **last_change**: 9 bits
  - Represent the key which has been pressed or released.



- **key_valid**: 1 bit
  - Should be active for one clock period (100MHz) when any key is pressed or released.

Verilog Module: KeyboardDecoder (4/5)

- Use an array "**key_down**" of 512 bits to record which key is currently being pressed
  - 1 means "key is pressed", while 0 means "key is released"



- The key indexed by "**0_0100_0000**" is pressed
- The key indexed by "**0_0100_0001**" is released

- Use "**key_decode**" (512 bits) to represent the key that was just pressed or released

Verilog Module: KeyboardDecoder (5/5)

- To represent that a key is pressed
  - key_down <= key_down | key_decode;

	0	1	1	0	1
or	0	0	0	1	0
	0	1	1	1	1

- To represent that a key is released
  - key_down <= key_down & (~key_decode)

	0	1	1	0	1
and	1	1	0	1	1
	0	1	0	0	1

Verilog Module: SampleDisplay

- In Keyboard Sample Code
  - KeyboardDecoder.v
  - SampleDisplay.v

- I/O for KeyboardDecoder

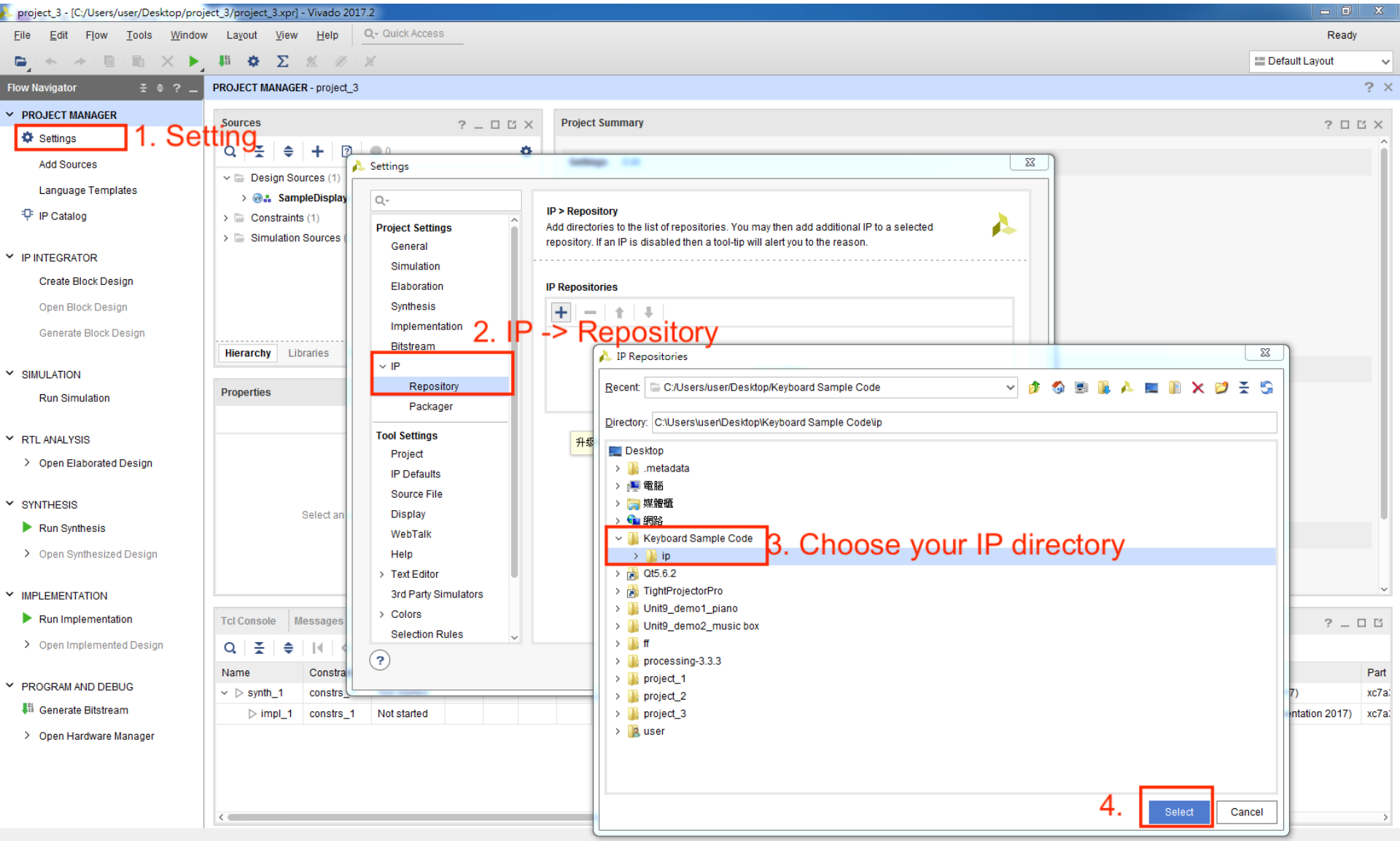
Output

- display
- digit

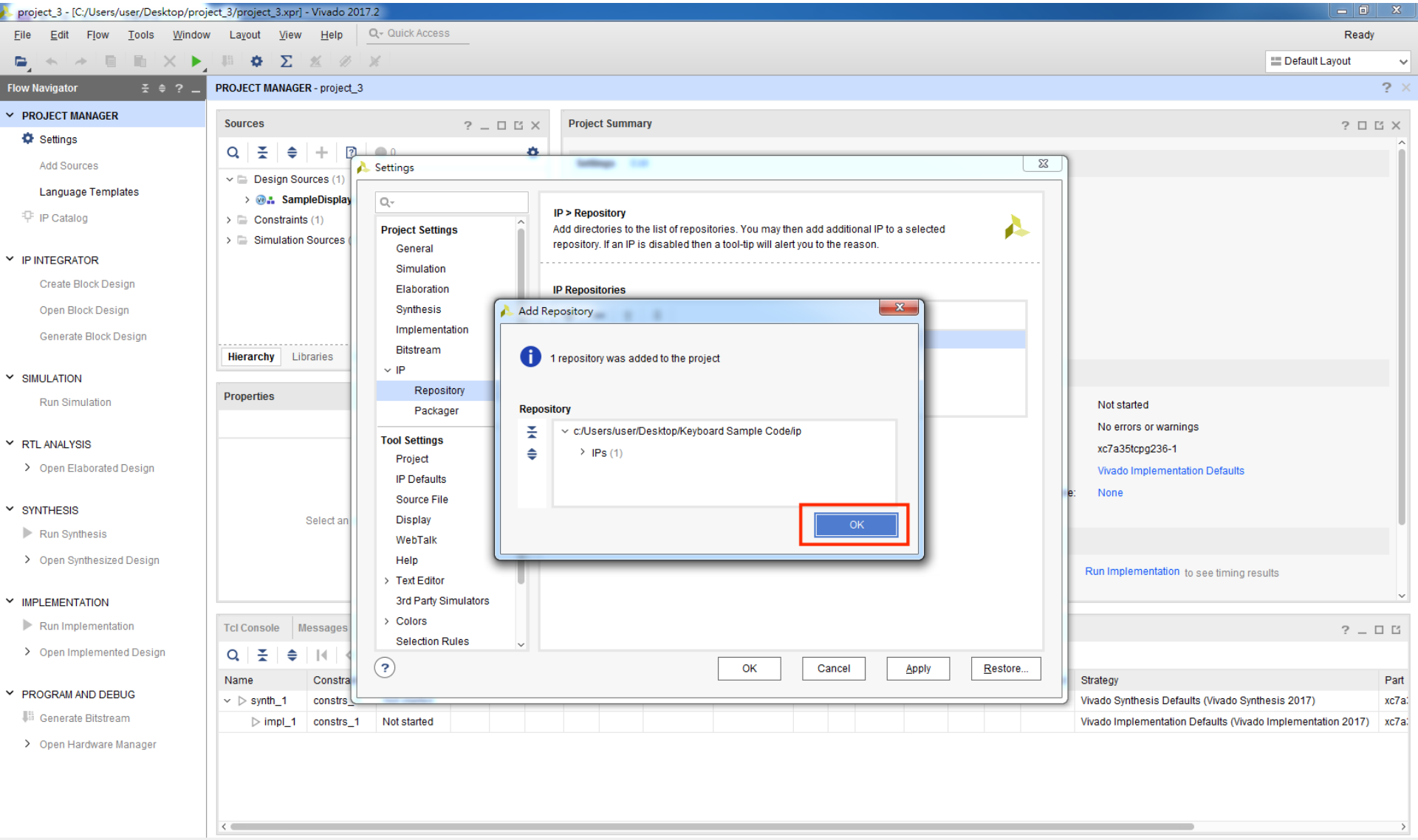
Input

- PS2_CLK
- PS2_DATA
- rst
- clk

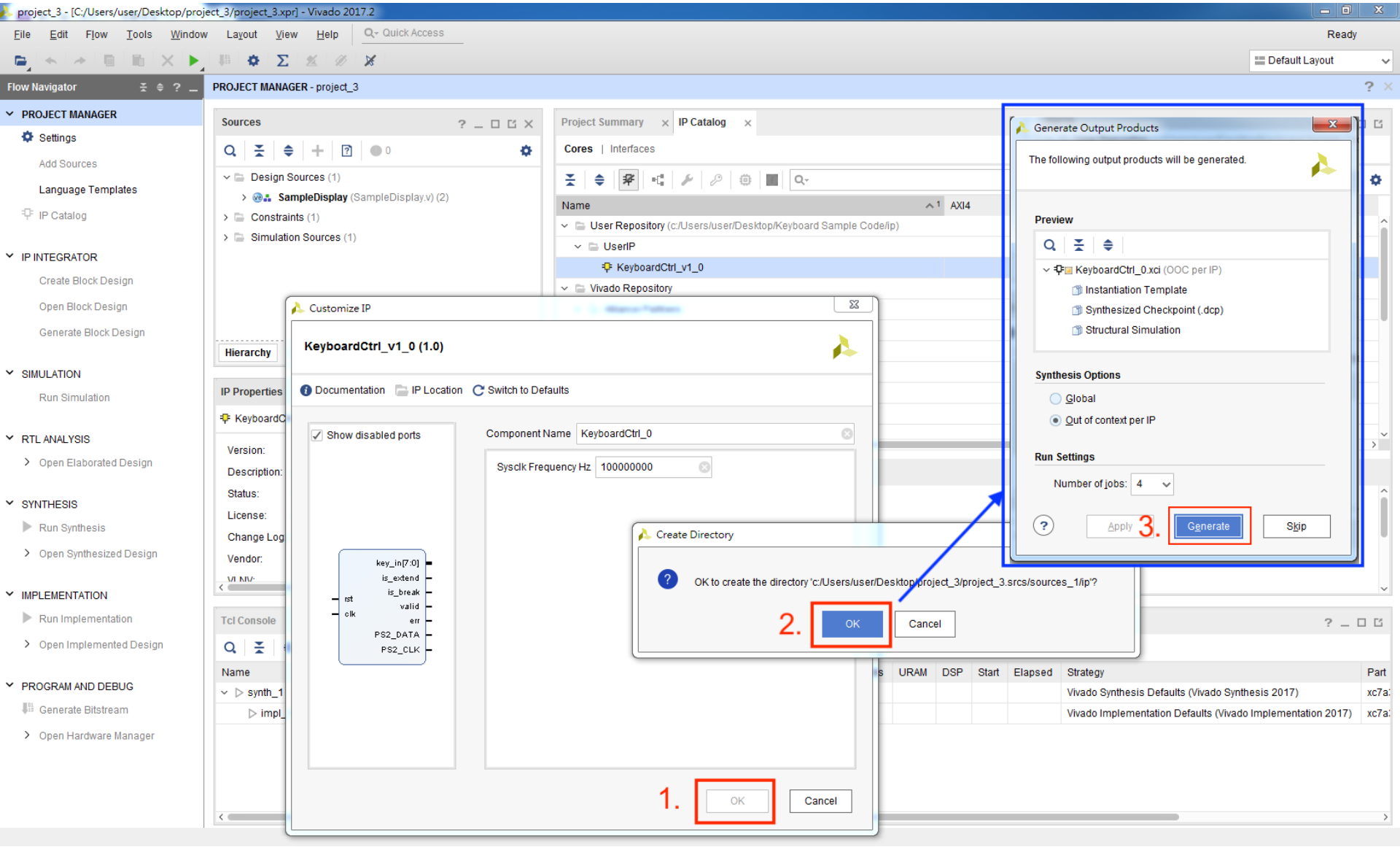
How to Use IP (1/4)



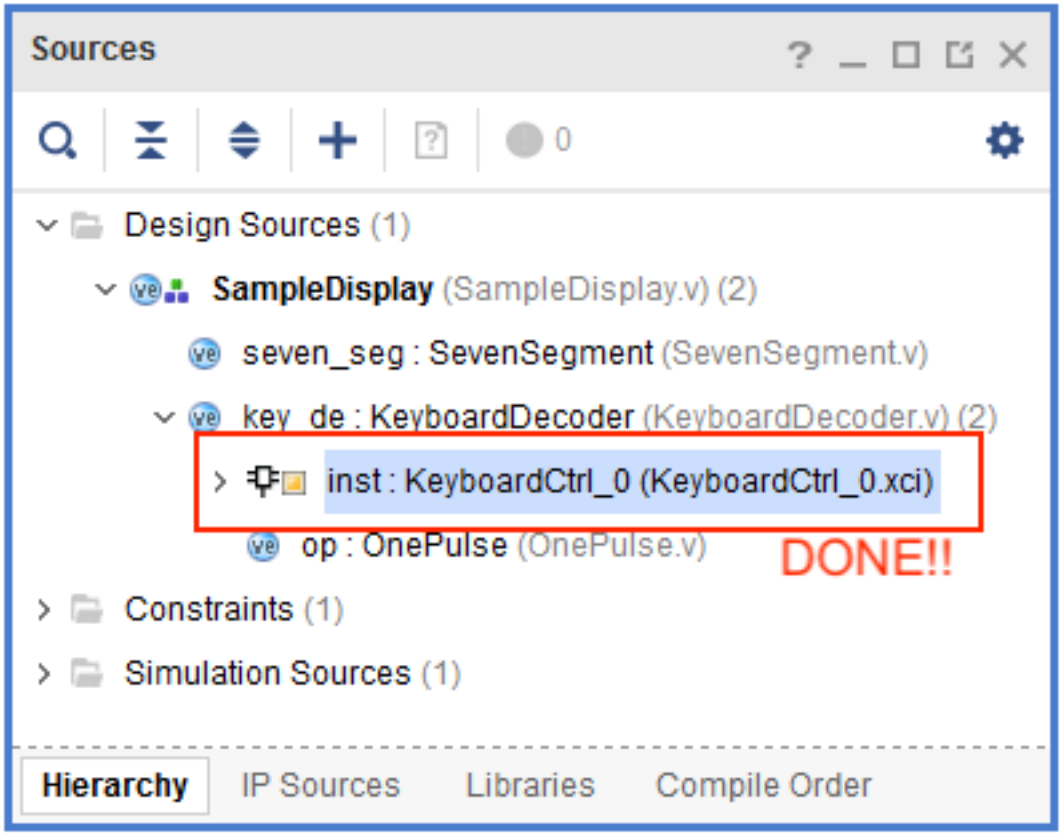
How to Use IP (2/4)



How to Use IP (3/4)



How to Use IP (4/4)



Picture was taken by Chun-Yi Lee himself, who was an amateur photographer