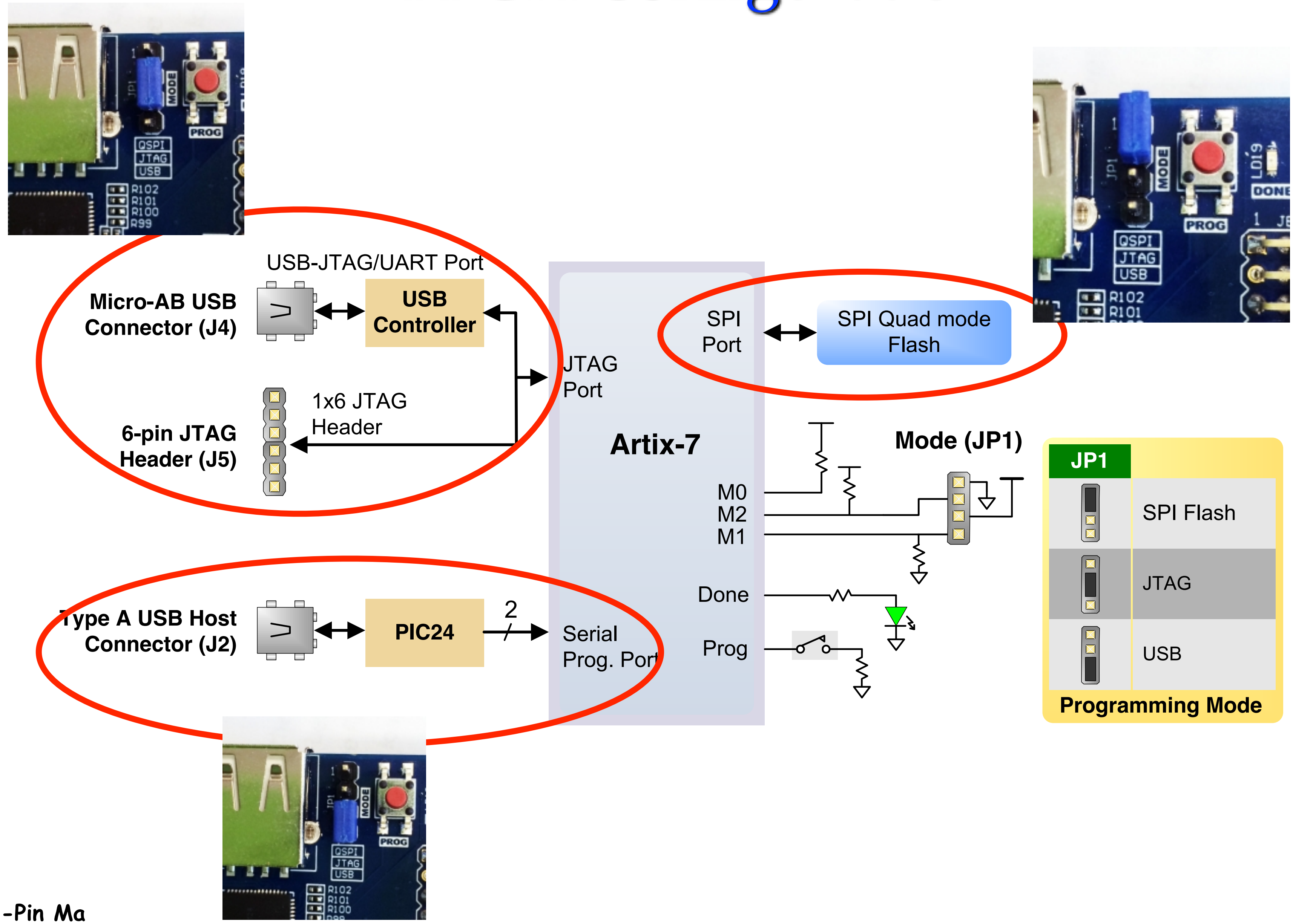
The Keyboard

Hsi-Pin Ma

https://eeclass.nthu.edu.tw/course/18498

Department of Electrical Engineering
National Tsing Hua University

FPGA Configuration

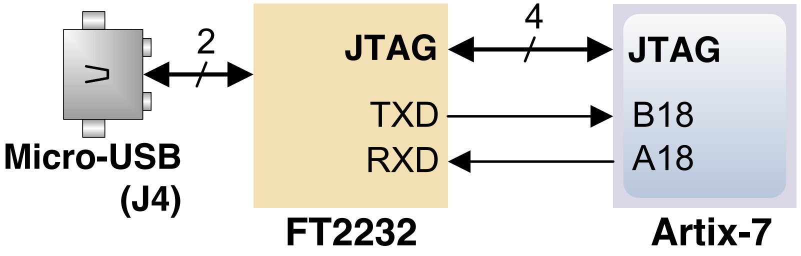


JP1	
	SPI Flash
	JTAG
	USB

Programming Mode

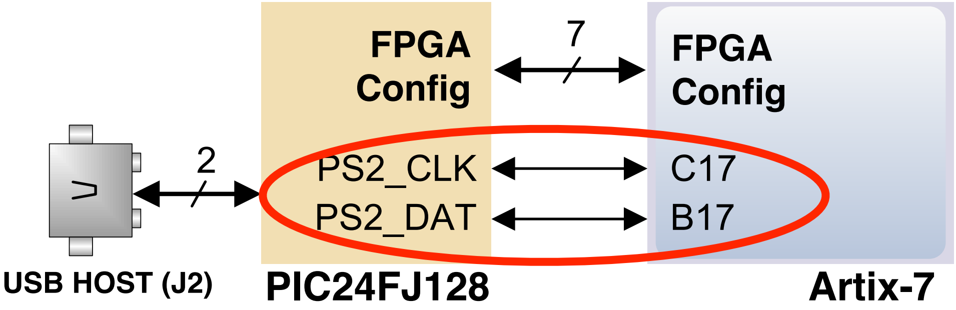
USB-UART Bridge (Serial Port)

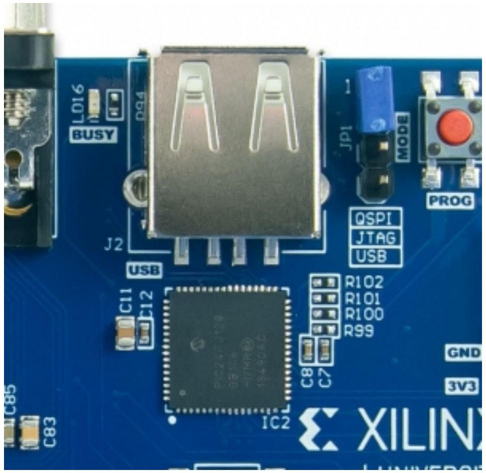
- FTDI FT2232HQ
  - Allow to use PC applications to communicate with the board using standard Windows COM port commands
    - Virtual Com Port convert USB packets to UART/serial port data
    - Serial port data is exchanged with the FPGA using a two-wire serial port (TXD/RXD)



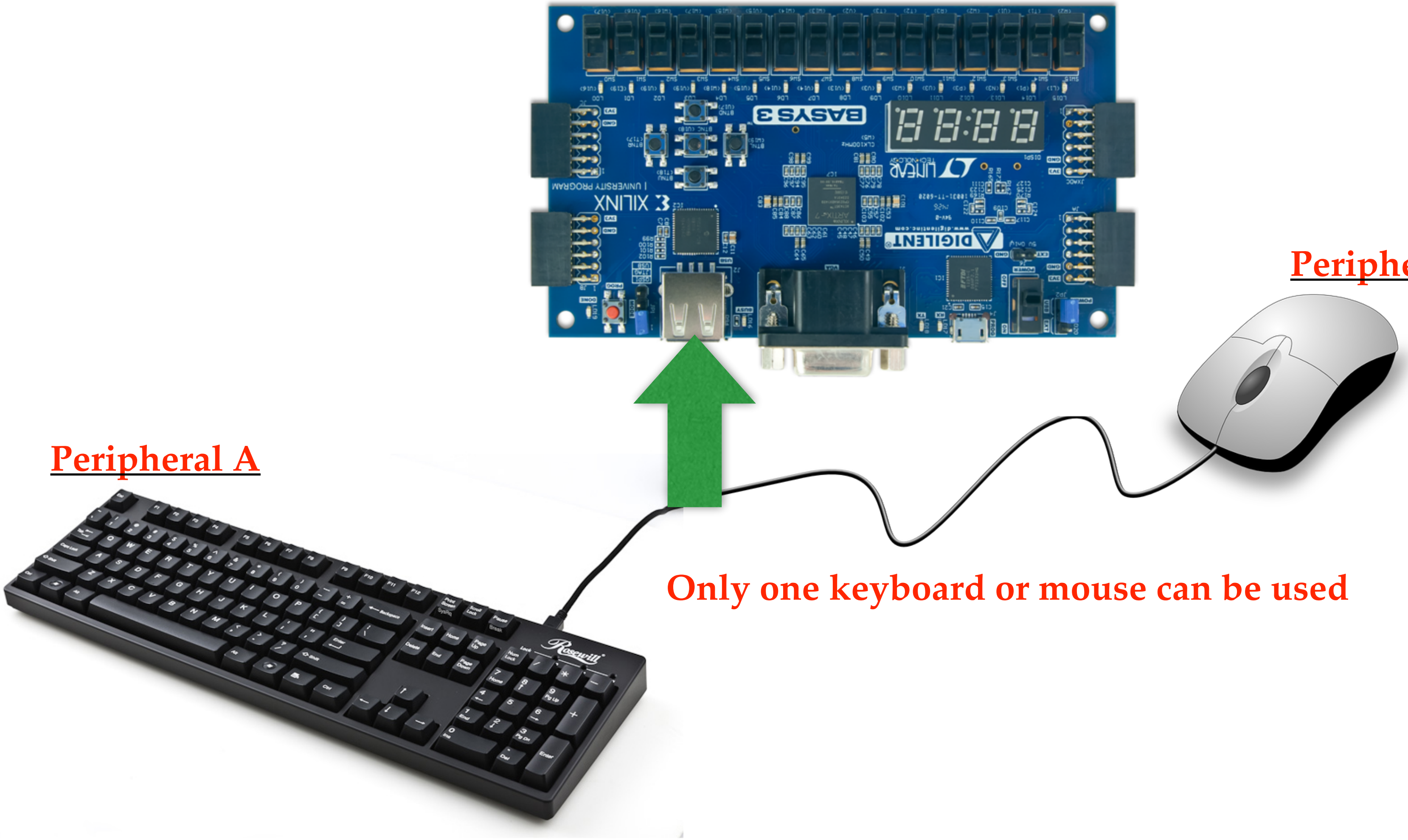
USB HID Host

- USB HID (Human Interface Device) host capability
  - Aux function microcontroller (Microchip PIC24FJ128)
- PIC24FJ128 function
  - Power-up: configuration mode
  - After programmed: application mode (USB HID Host mode)
  - Do not support Hub. Only a single mouse or a single keyboard can be used (standard PS/2 interface)





USB HID Host



Host device

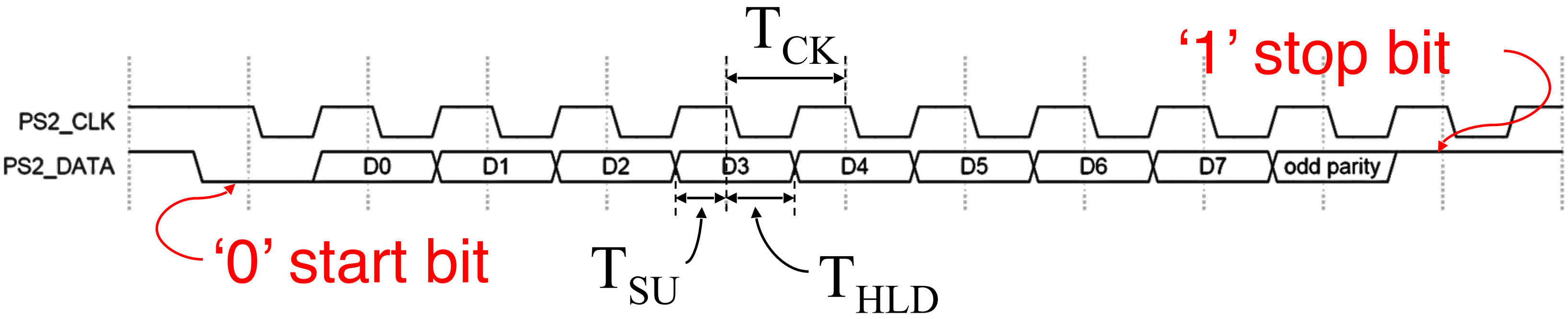
Peripheral B

Peripheral A

Only one keyboard or mouse can be used

HID Controller

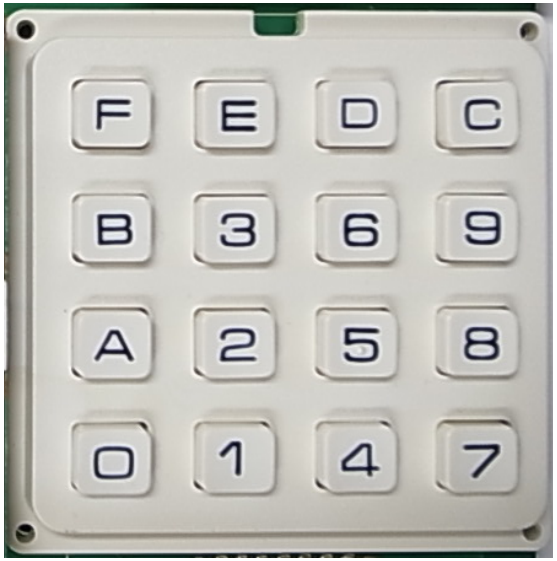
- 11-bit words protocol
  - start bit, data byte (LSB first), odd parity, stop bit

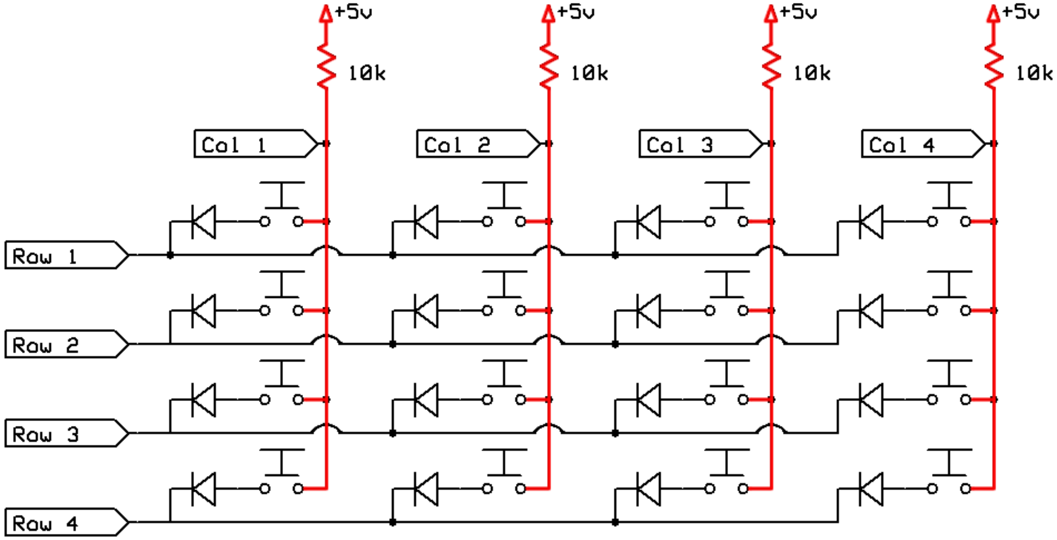


Symbol	Parameter	Min	Max
T_{CK}	Clock time	30us	50us
T_{SU}	Data-to-clock setup time	5us	25us
T_{HLD}	Clock-to-data hold time	5us	25us

4x4 Keypad

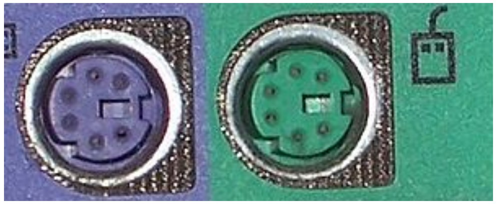
- Keypad scan

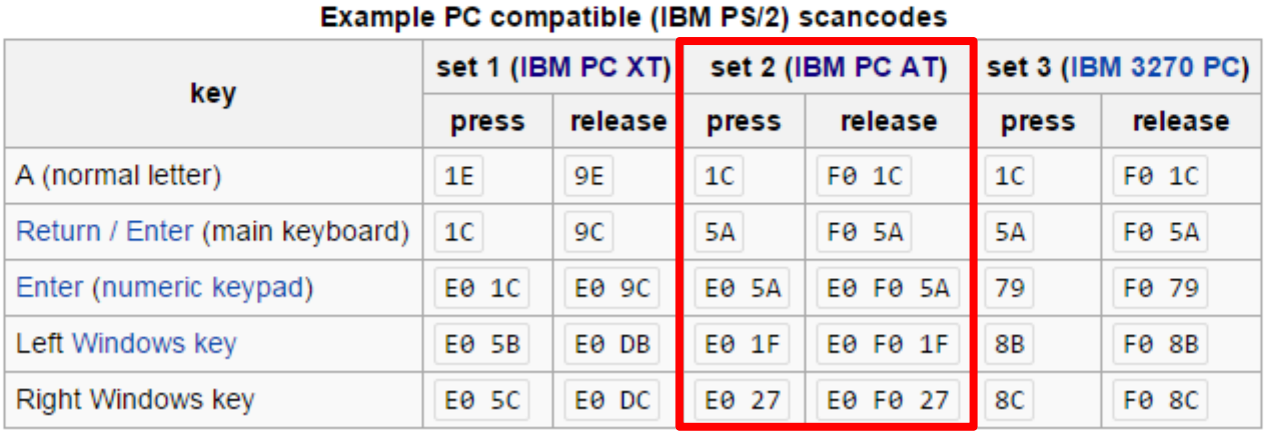




PS/2-style Keyboard Operation

- Use *scan codes* to communicate key press data
  - Each key is assigned a code
  - If the key is held down, the scan code will be sent repeatedly about once every 100ms. (**make** code)
  - When a key is released, an **F0** key-up code is sent, followed by the scan code of the released key. (**break** code)
  - Some keys (right Ctrl, right Alt, …) , called extended keys, send an **E0** ahead of the scan code. (**extend** code)



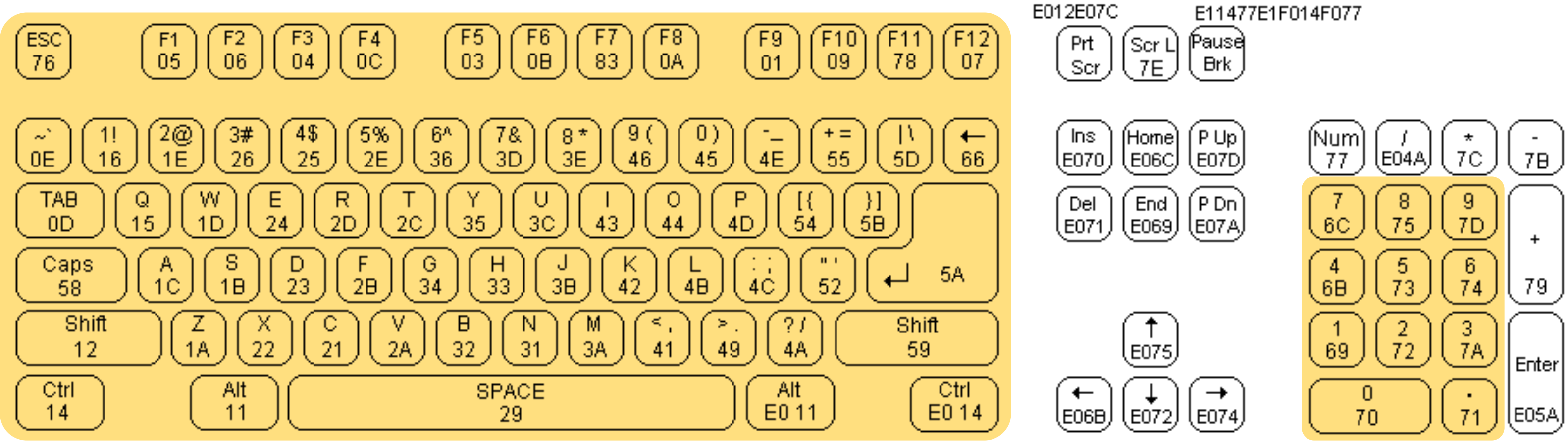


Example PC compatible (IBM PS/2) scancodes

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

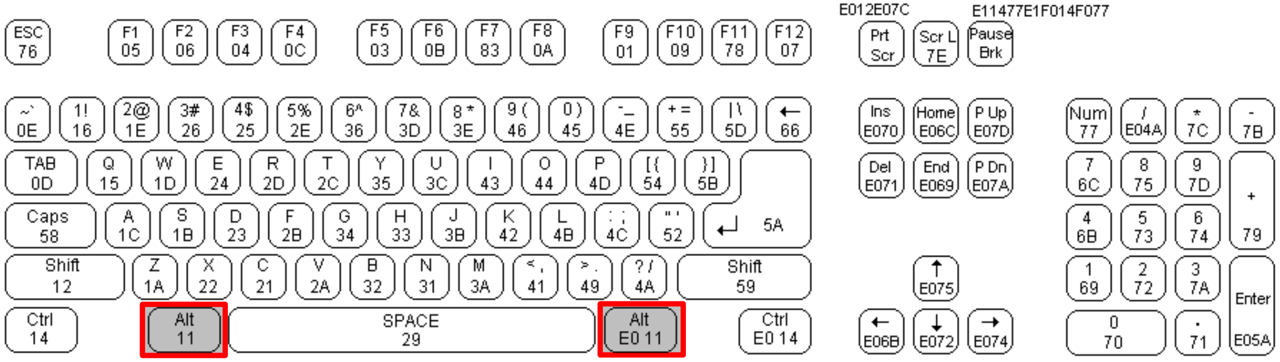
PS/2 Keyboard Scan Code



We only use the yellow parts of the keyboard.

Extend Code	Break Code	Make code
E0	F0	XX

(means "release")

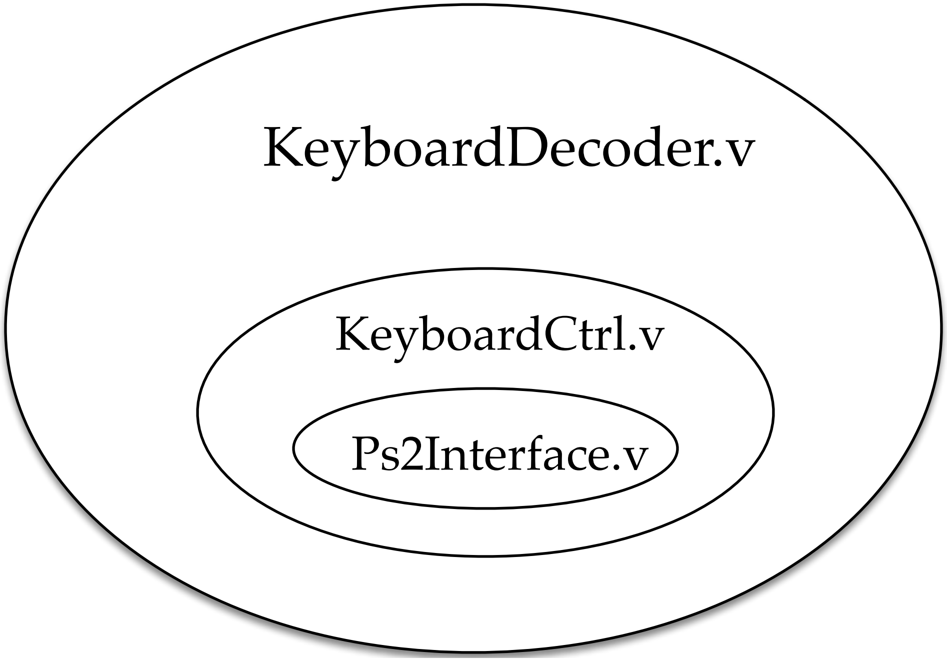


L Alt press			11
L Alt release		F0	11
R Alt press	E0		11
R Alt release	E0	F0	11

PS/2-style Keyboard Operation

- A host device can also send data to the keyboard
- The keyboard can send data to the host only when both the data and clock lines are high (or idle)
  - If the host drive the clock line low, the keyboard must not send any data until the clock is released.
- The keyboard generates 11 clock transitions (at 20 to 30 KHz) when the data is sent, and data is ***valid on the falling edge*** of the clock.
- When a keyboard or a mouse is connected to the Basys3, a "self-test passed" command (0xAA) is sent to Basys3
- A Read ID command (0xF2) from Basys 3 is used to identify what kind of device is connected

KeyboardDecoder

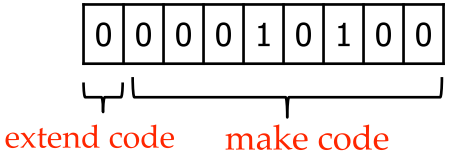


Inout:

PS2_DATA, PS2_CLK

rst (high active reset)

clk (100MHz)



0	0	0	0	1	0	1	0	0

Outputs:

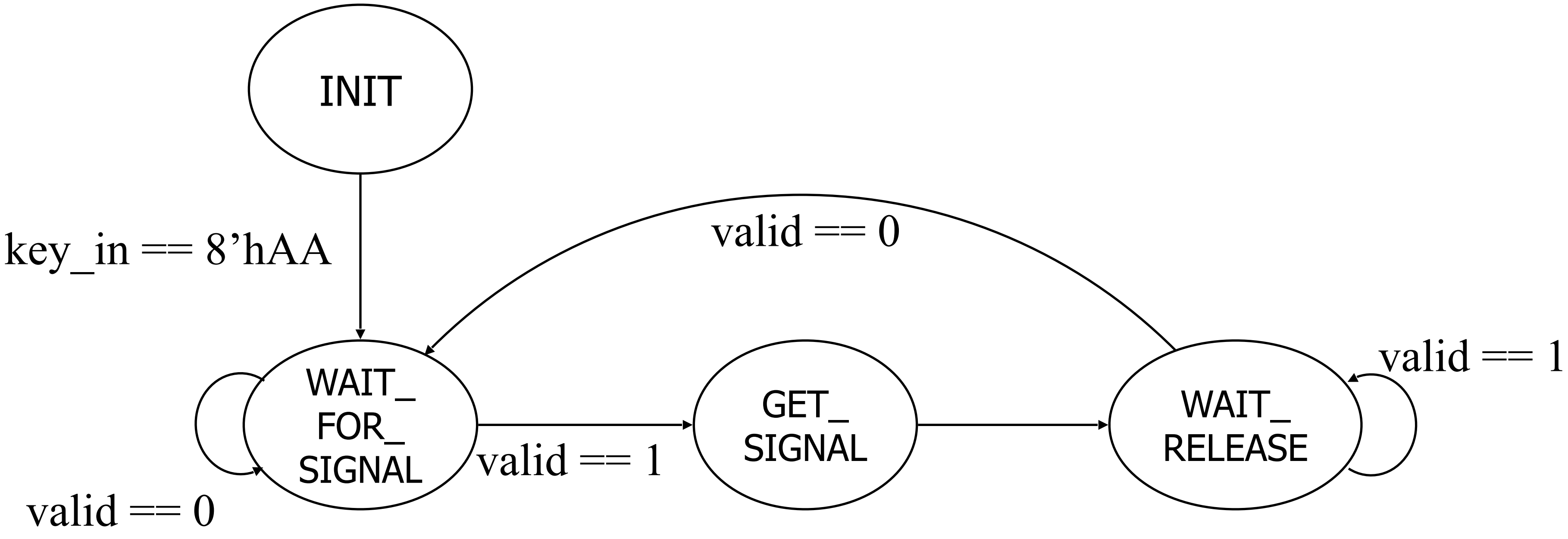
last_change [8:0]

key_down [511:0]: address of the key pressed

key_valid: high for 1 clock period (100MHz) when a key is pressed or released

KeyboardDecoder

- FSM in the decoder



For Lab8

- Use KeyboardDecoder to get the codes of pressed key
- Remember to add three .v files into your projects
  - Ps2Interface.v
  - KeyboardCtrl.v
  - KeyboardDecoder.v