# Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: ⭘ M1 ⭘ T1

# Homework Assignment: Submit via gradescope

1. In this problem, you will build a VHDL entity called **scancode decoder** which processes keyboard scancodes. When you press a key on a keyboard, the keyboard sends an 8-bit code to the computer called a PS2 scancode. Each key has its own scancode listed below. The relationship between the keys and their scancode is not based on ASCII nor any other discernible pattern.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyboard Key | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Scancode (in hex) | 0x45 | 0x16 | 0x1E | 0x26 | 0x25 | 0x2E | 0x36 | 0x3D | 0x3E | 0x46 |

Build a component which converts an 8-bit scancode for the digits 0-9 into a 4-bit hexadecimal values.

|  |  |
| --- | --- |
| Nomenclature: | scancode\_decoder |
| Data Input: | scancode = std\_logic\_vector(7 downto 0); |
| Data Output: | decoded\_value = std\_logic\_vector(3 downto ); |
| Control: | none |
| Status: | none |
| Behavior: | Converts scancode, representing a keypress of a decimal digit, into its 4-bit value. For example, if scancode = 2516, the scancode for the character "4", then the converter should output decoded\_value = 01002. Assume that the inputs are always legal hexadecimal scancodes. |

Use the ***when*** statement syntax to describe the output in terms of the input.

1. Create a testbench to simulate and demonstrate your function works for all 10 values in the first table. Provide a cropped screenshot of the resulting simulation timing diagram below - make the signal values legible.

**PLACE DIAGRAM HERE**

1. In Github, provide the VHDL code for your **Scancode decoder** component and testbench.