Embedded Systems Course – General Purpose I/O (GPIO)

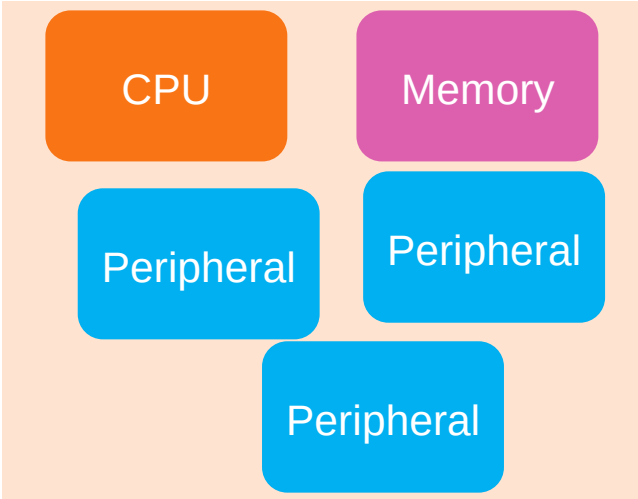
Prakash Balasubramanian

8 Nov 2024

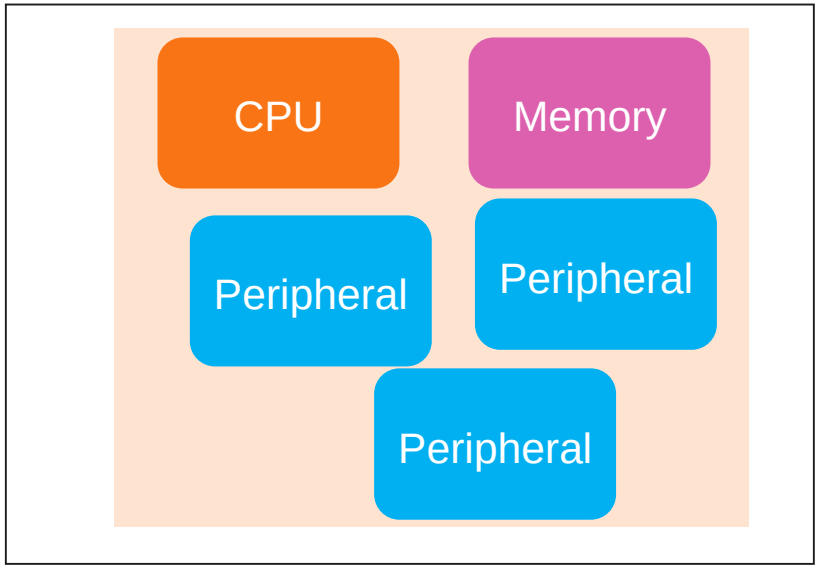
Pins, Pads, Ports – 1/3

This is a Silicon

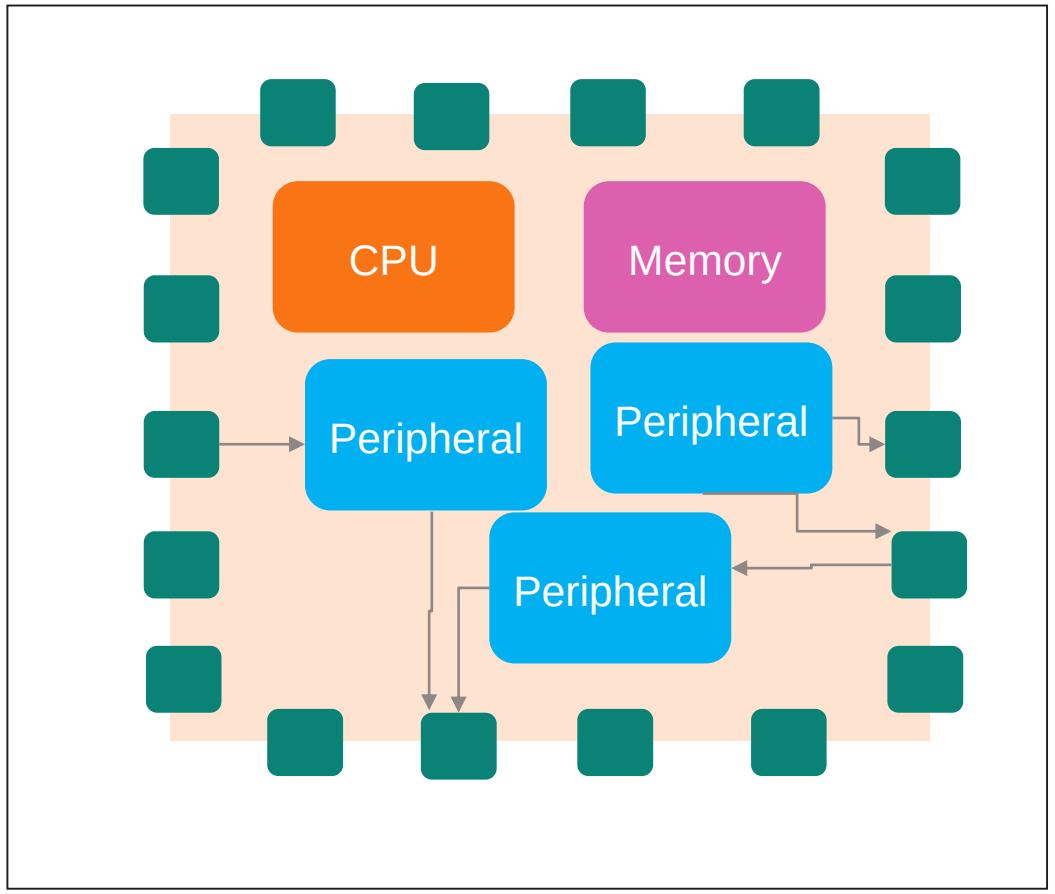
This is silicon with CPU, Memory and peripherals



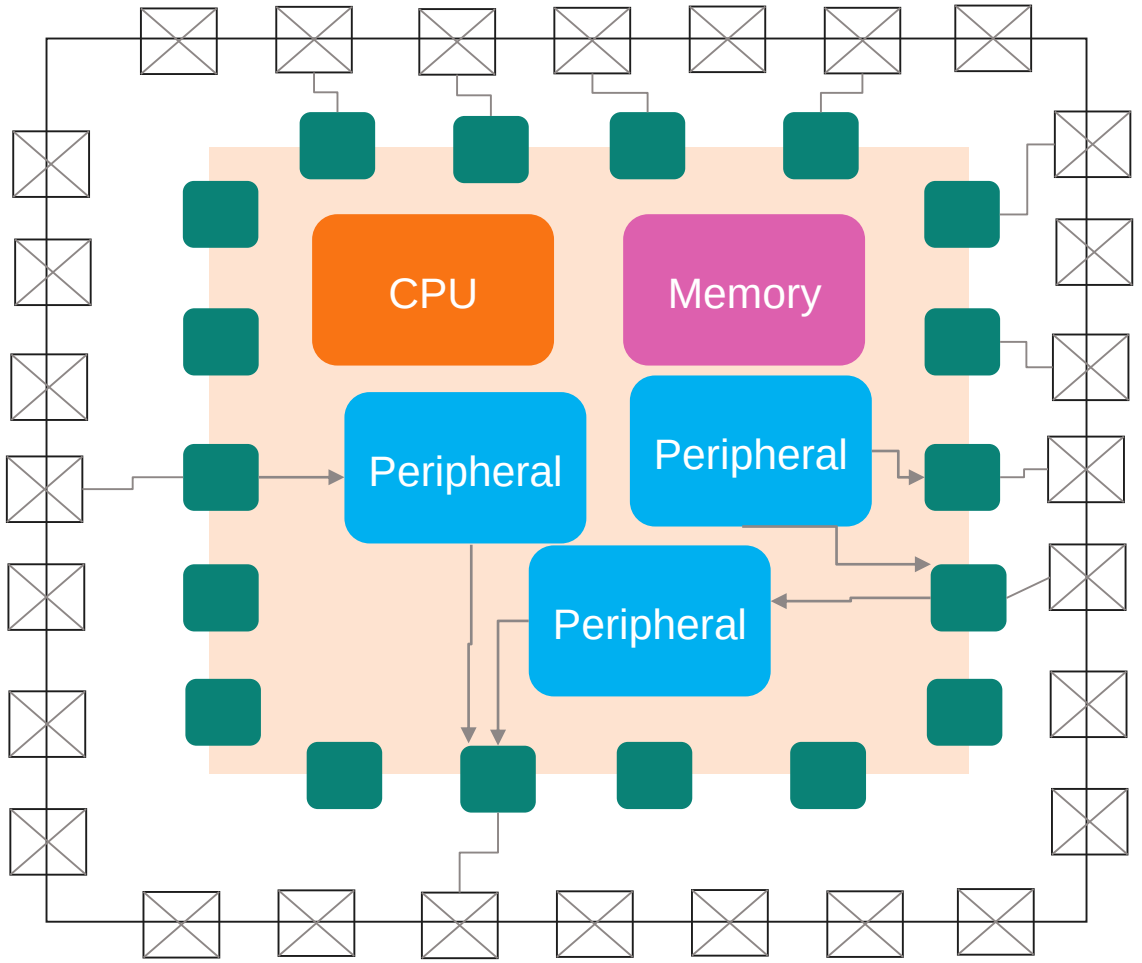
This silicon is placed inside a “Package”



Pins, Pads, Ports – 2/3

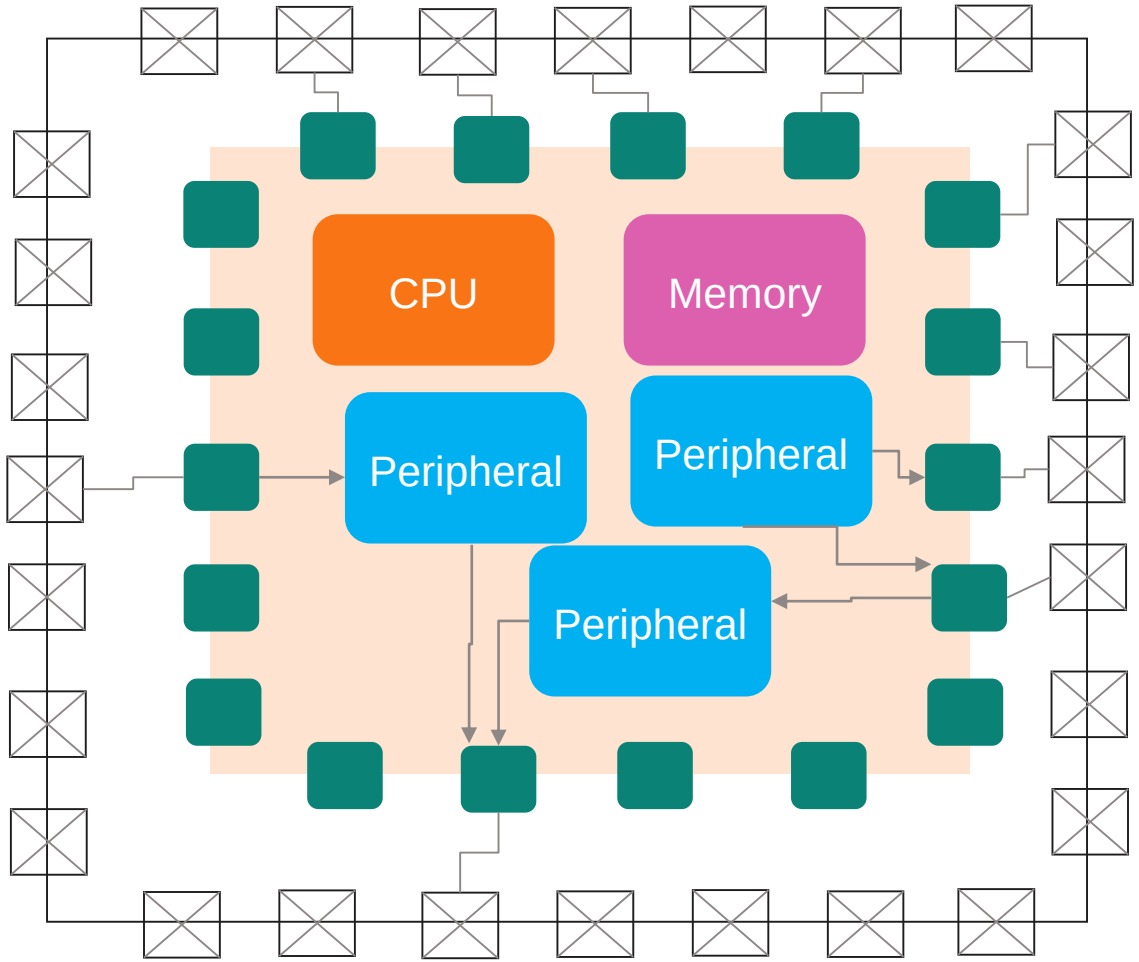


External world signals enter and exit the silicon through pads



Pads are connected to package pins by wires (typically gold)

Pins, Pads and Ports – 3/3

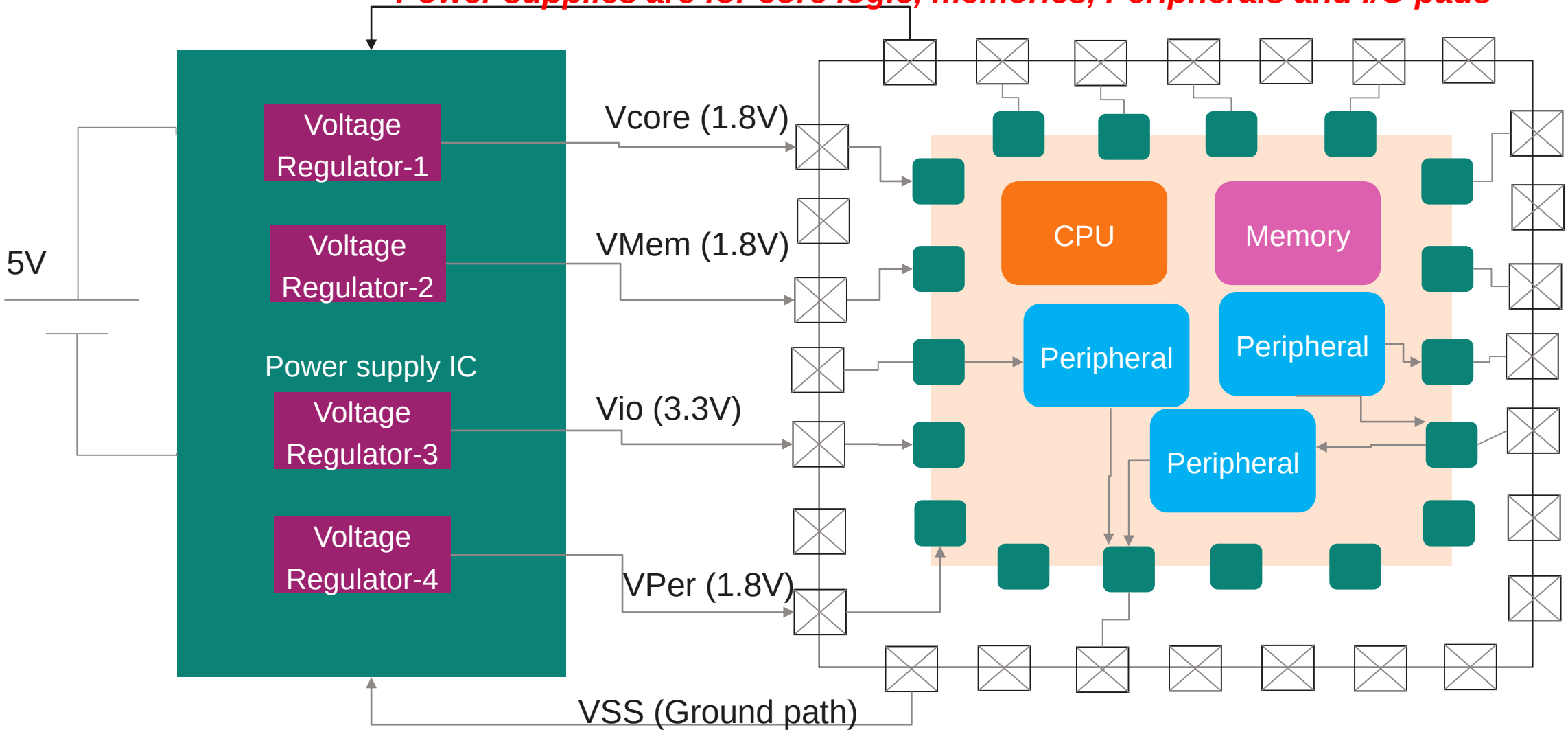


- The number of package pins may be greater than pads available on silicon
- A pad may be connected to signals from multiple peripherals
- Some package pins are connected to power supplies and their ground lines
- Not all package pins are necessarily connected to pads

Power supplies

Without the 4 power supplies, the chip just will not work!!

Power supplies are for core logic, memories, Peripherals and I/O pads



Pads

Communication between MCU (Microcontroller) and external world is ONLY via pads

Pads are like gateways

A pad can be an input pad (i.e. a signal from a sensor can enter the chip)

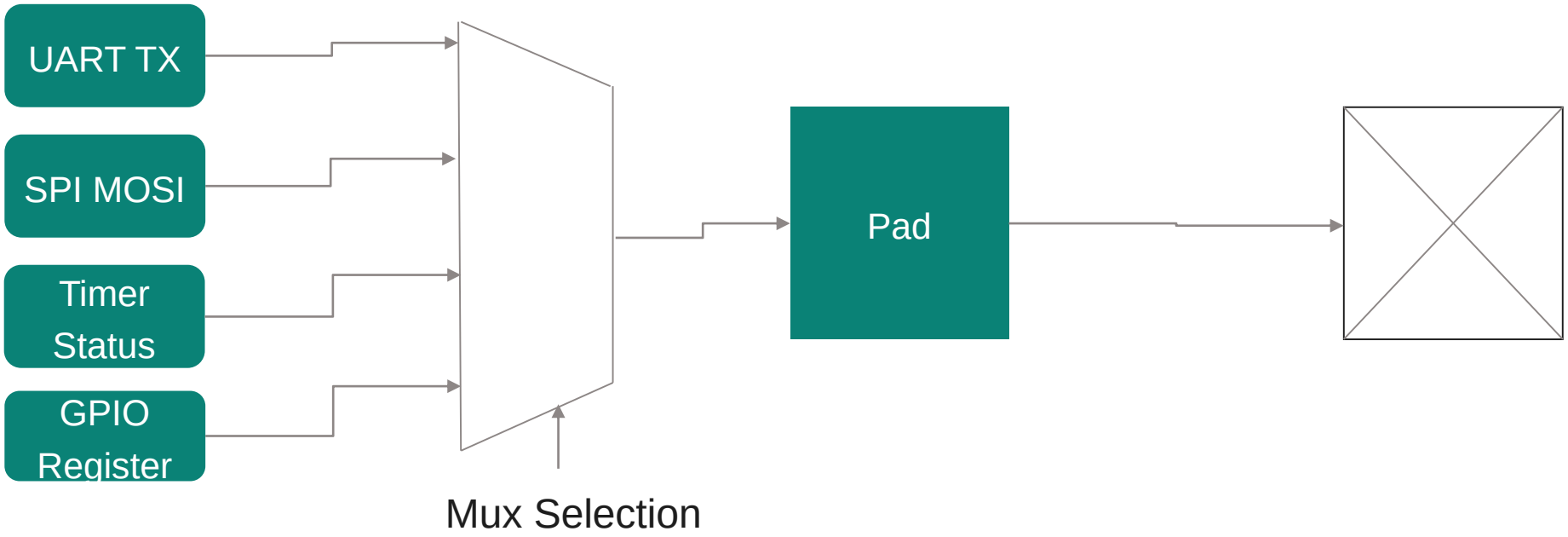
A pad can be an output pad too (i.e. Signals are sent out of the chip)

A signal from a pad can either be connected to a peripheral

Or that pad signal can be delivered to/controlled by software

This is the famous GPIO (General Purpose I/O) functionality

Pads – 1/3



In this example, the pad (an output pad in this case) is connected to 3 peripherals

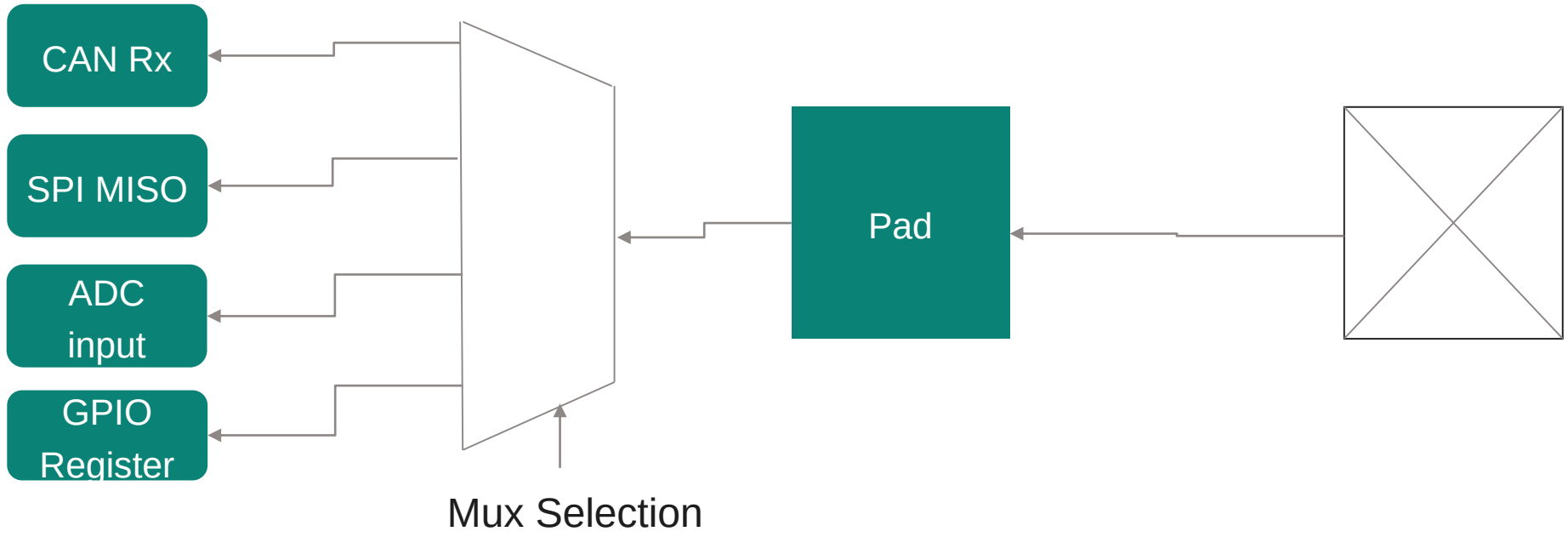
This pad is also connected to a register which can be written by your software

Among the 4 options available, you as an application developer must decide what should this pad be used as!

You do so with the help of a mux selector (More on this later)

The 4 options of a pad are described in the user manual of the MCU

Pads – 2/3



You may be tempted to think that all pad options have the same direction

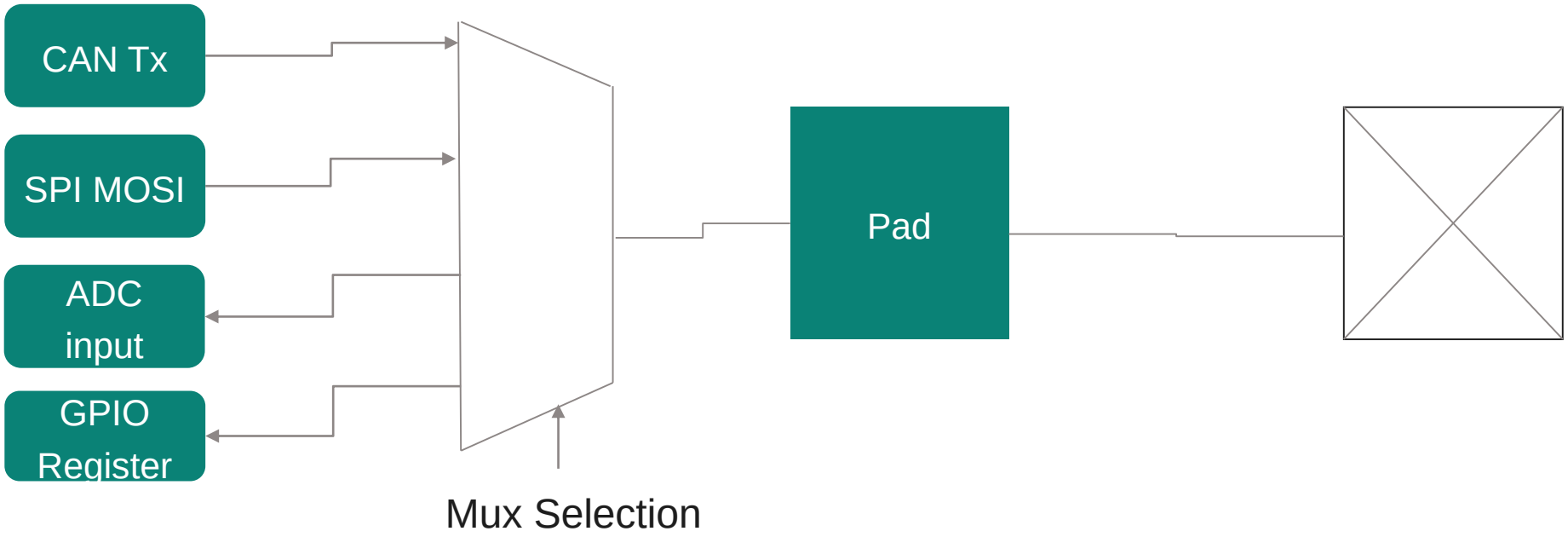
That simply may not be true

In this picture above, all 4 options are depicted as inputs

Such an arrangement is rare

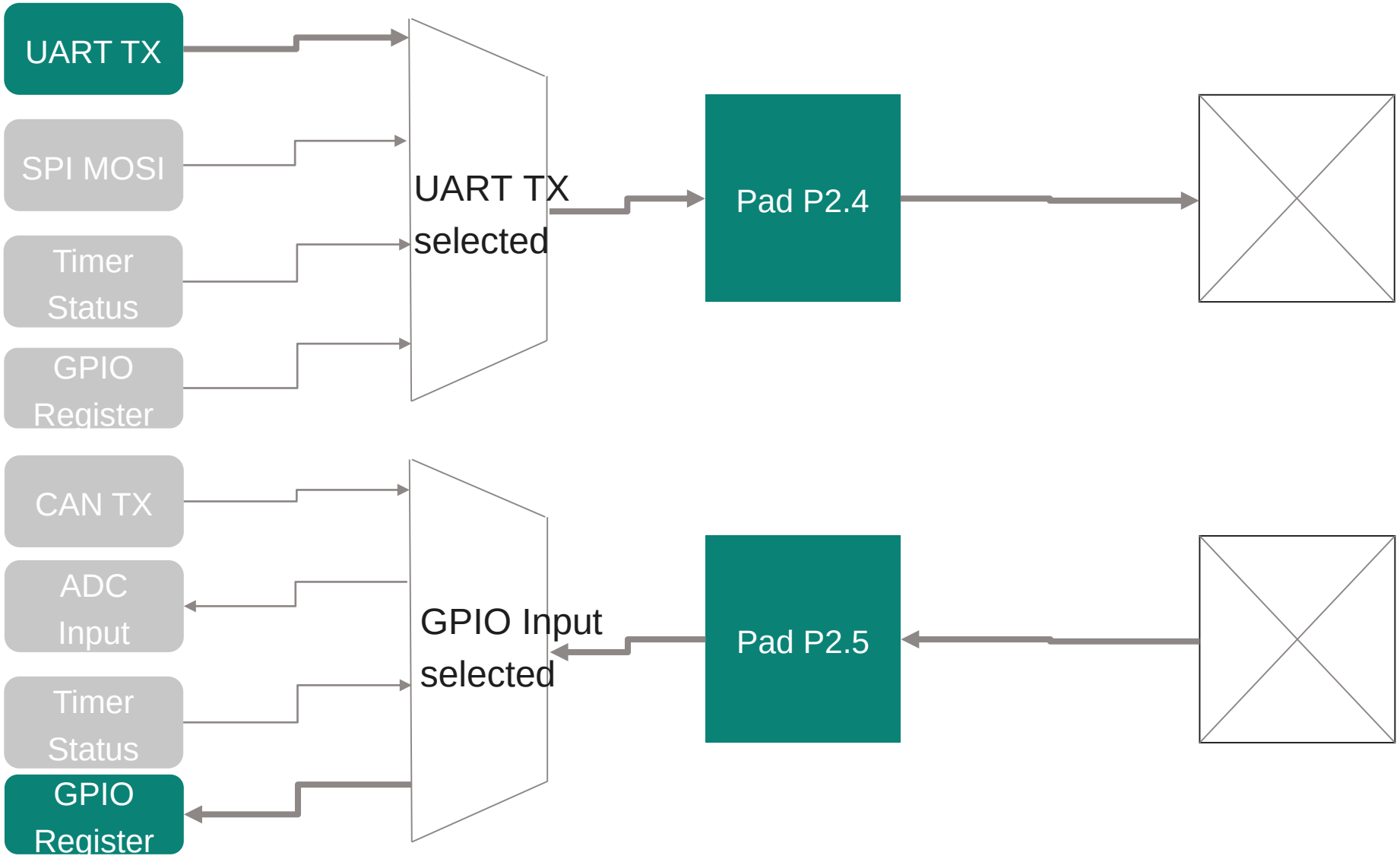
But you get the main idea about options and your obligation to choose one as per your needs

Pads – 3/3

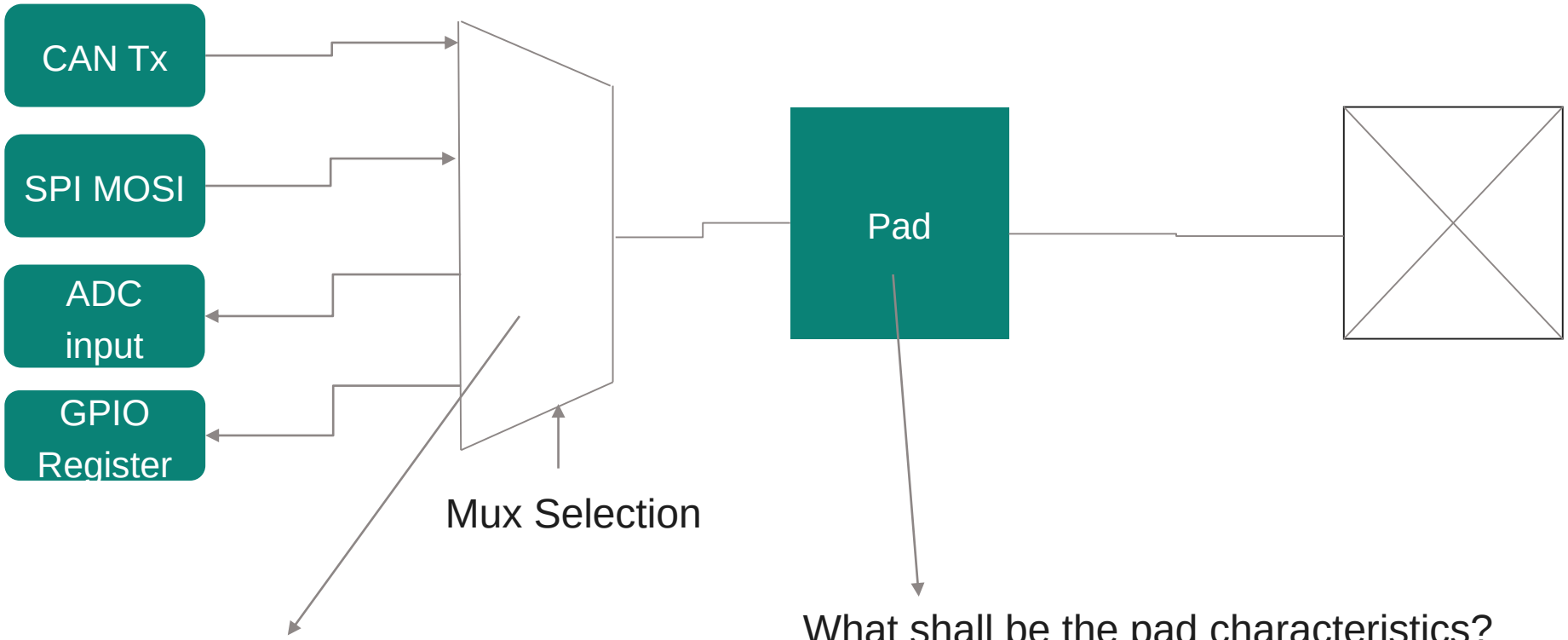


In reality, the chip architect may provide both input and output signaling capability.
It is for you to decide how this pad must be configured and used.

An example



Basics of pad configuration – 1/5

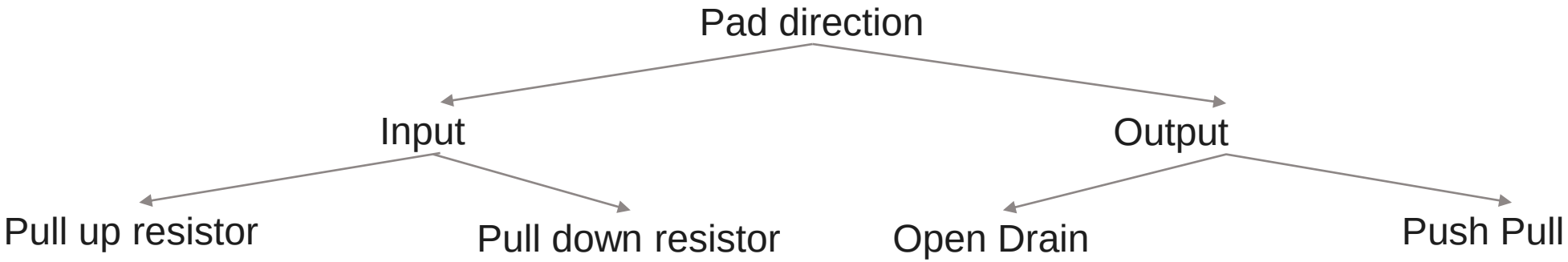


Which of the many options should be selected?

What shall be the pad characteristics?

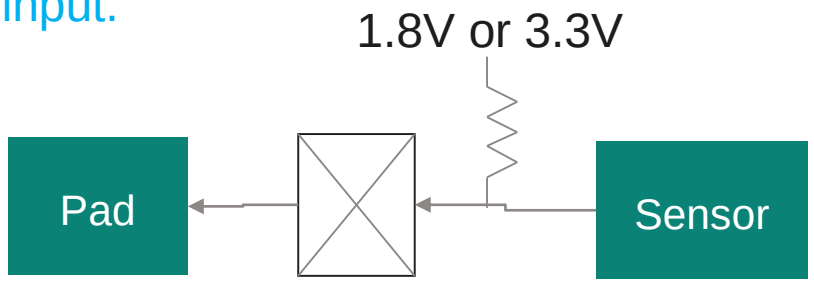
- Pad direction
- Pad strength
- Output options [Open Drain, Push Pull]
- Input options [Pull Up, Pull Down]

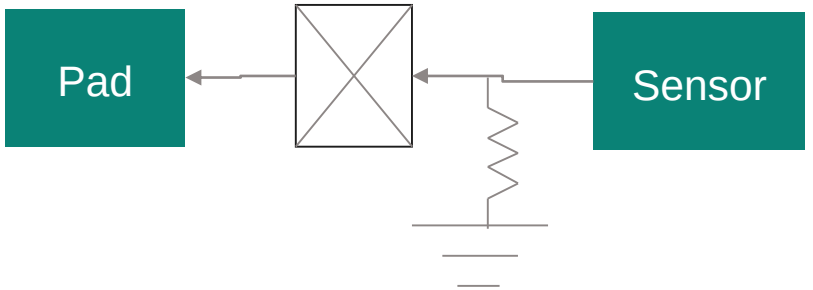
Basics of pad configuration – 2/5



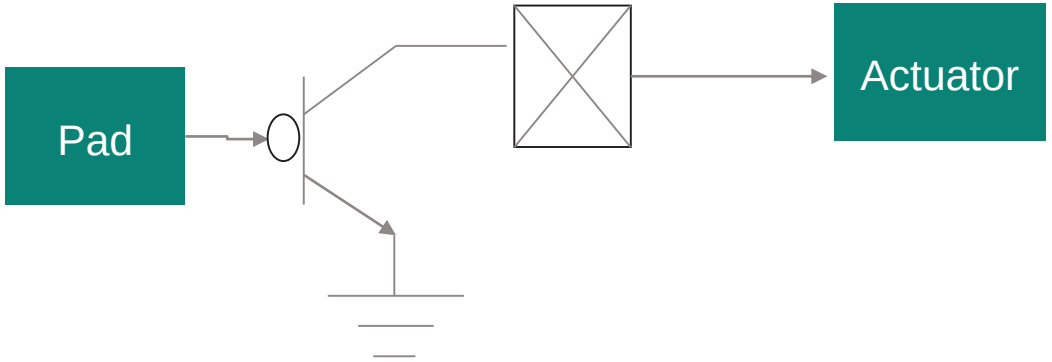
Basics of pad configuration – 3/5

When the pad is configured as an input pad, be sure to pull up or pull down the input line to suppress noise. **When the sensor does not drive any input** and **if you have connected a pull-up resistor**, the voltage appearing at pad input is the 1.8V or 3.3V. **If instead you connected a pull-down resistor**, then 0V appears at pad input.



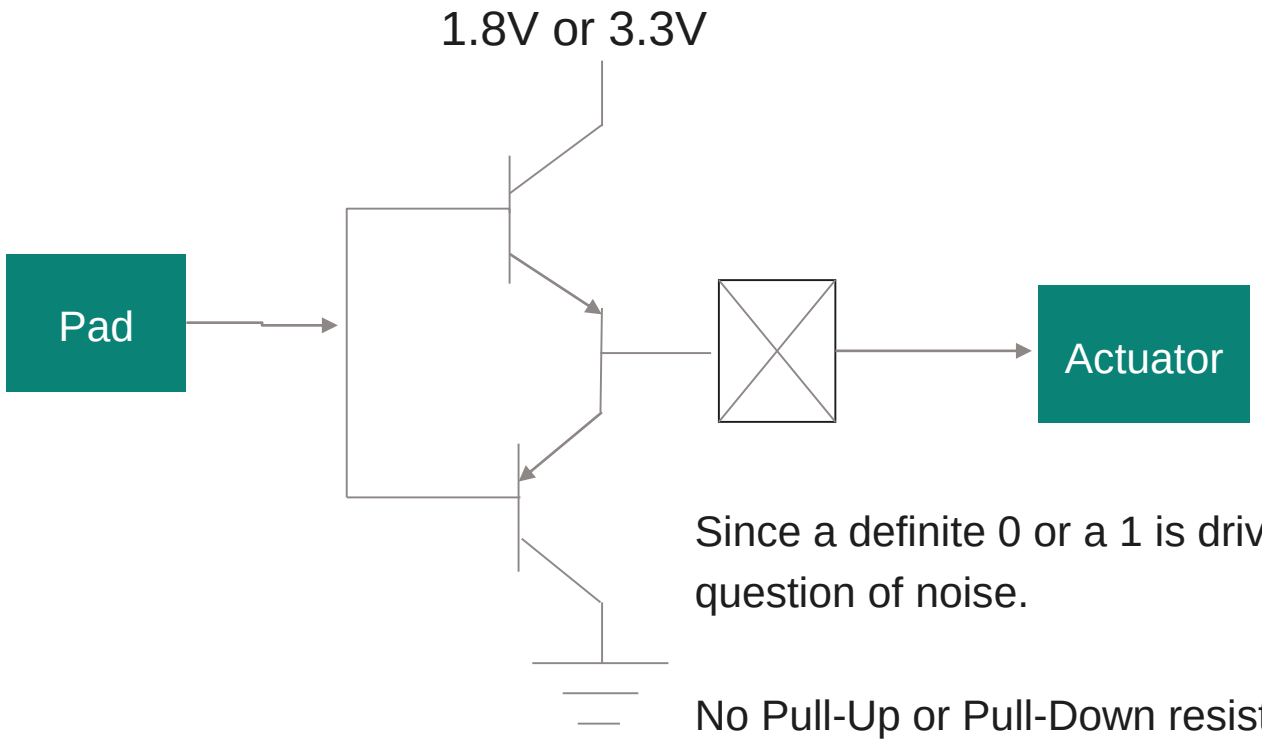


This pad below is configured as output and set to Open-Drain mode. When the pad output is a 0, the transistor will conduct and pull the output line to actuator to 0. When the pad output is a 1, the transistor does not conduct and the output is floating. So, you must connect an external Pull-Up or Pull-Down resistor to suppress noise



Basics of pad configuration – 4/5

This pad below is configured as output and set to push-pull mode. When the pad output is a logic-0, the PNP transistor will conduct and pull the output line to actuator to logic-0. When the pad output is a logic-1, the NPN transistor on top conducts and a logic-1 appears on the actuator line.



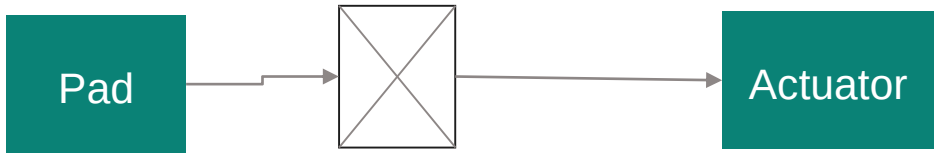
Since a definite 0 or a 1 is driven on the line by the transistors, there is no question of noise.

No Pull-Up or Pull-Down resistors are required.

Basics of pad configuration – 5/5

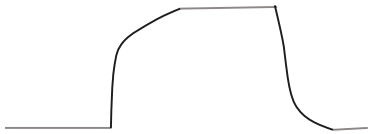
Pad drive strength

Although you do not see it, the path from pad to the output device such as the actuator shown in this picture is actually a transmission line. Transmission lines have capacitances which cause change in rise and fall times of the signal driven by the pad.



MCU transmitted this on the pad

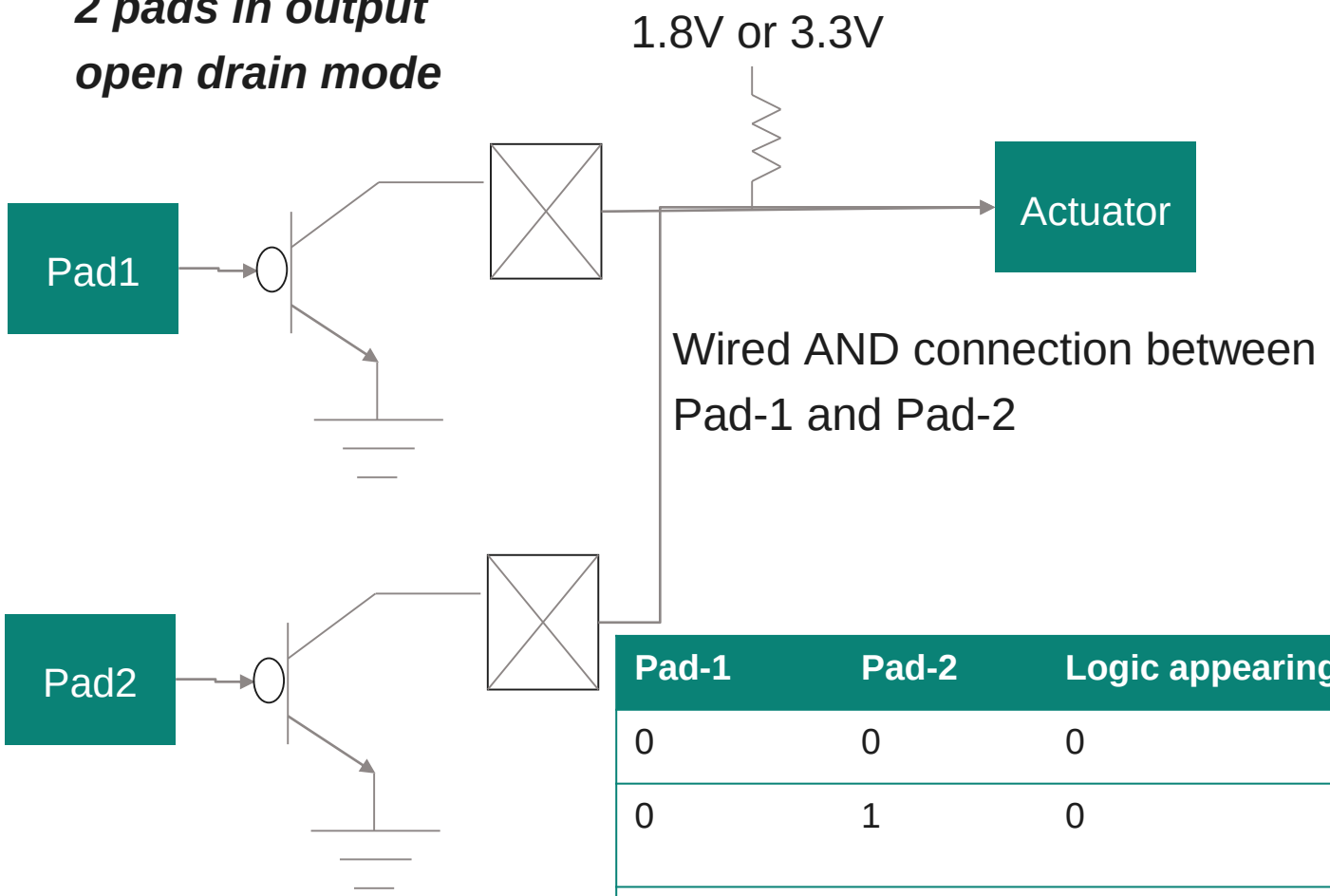
But the actuator ended up receiving this instead



By programming the drive strength (slew rate) of the pad, the capacitive effects of the transmission line can be mitigated and a good replica of the transmitted data is guaranteed to reach the destination.

When to use Open drain and when to use push-pull?

2 pads in output open drain mode



When only 1 pad drives an output device like the actuator, a Push-Pull may be appropriate.

But if one output line must be driven by multiple output pads, then an Open Drain with a pull up is definitely suitable.

Pad-1	Pad-2	Logic appearing at Actuator	Explanation
0	0	0	Both transistor conduct. Grounds are shorted
0	1	0	Pad-1 transistor conducts while Pad-2 transistor does not. Ground of Pad-1 appears at sensor input
1	1	1	Both transistors do not conduct and the pads are isolated from the actuator. The external supply gets connected to sensor through the Pull-Up resistor

How can a Pad be configured?

- Each pad has a set of software programmable registers
- As an example, one of the registers can be for configuring pad direction, pad multiplexer select, pad input or output choices
- There can be a register for writing desired output value when that pad has been configured as output
- There can be a register for reading pad input value when that pad has been configured as input

Pad-x configuration register	Pad-y configuration register	Pad-z configuration register
Pad-x output register	Pad-y output register	Pad-z output register
Pad-x input register	Pad-y input register	Pad-z input register

There are 3 pads Pad-x, Pad-y and Pad-z. For each pad, there are a set of user programmable registers. The names and number of such registers differ across MCUs. Please read the user manual of your MCU.

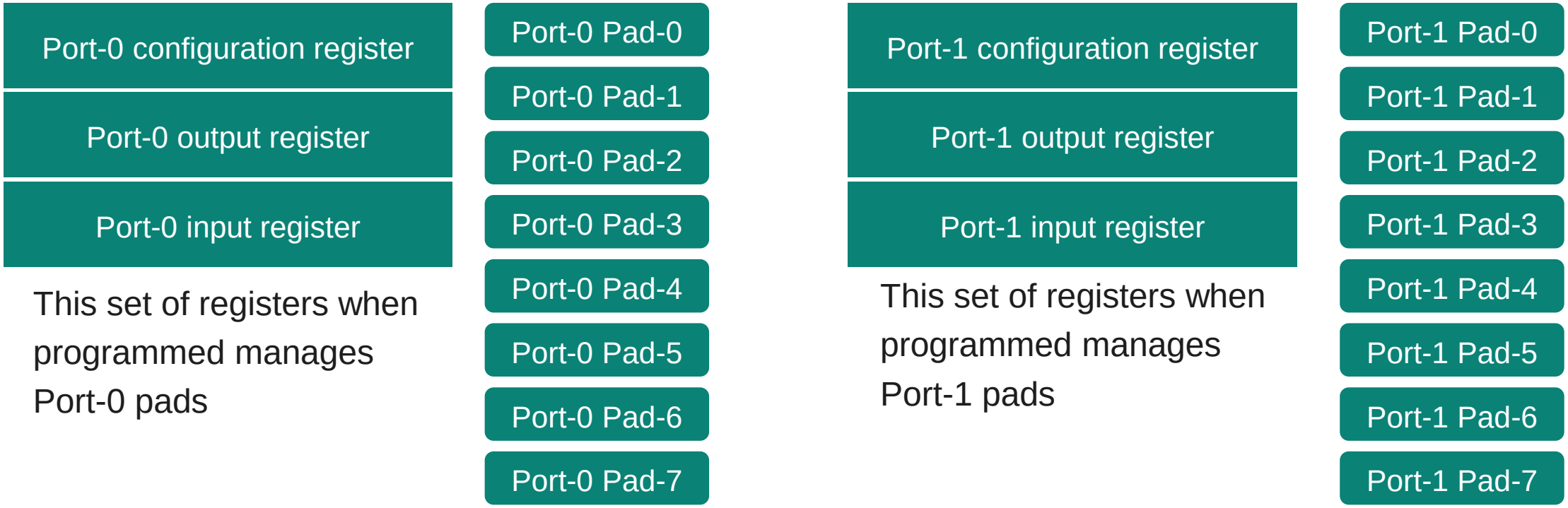
PORT – 1/2

A PORT is a collection of pads.

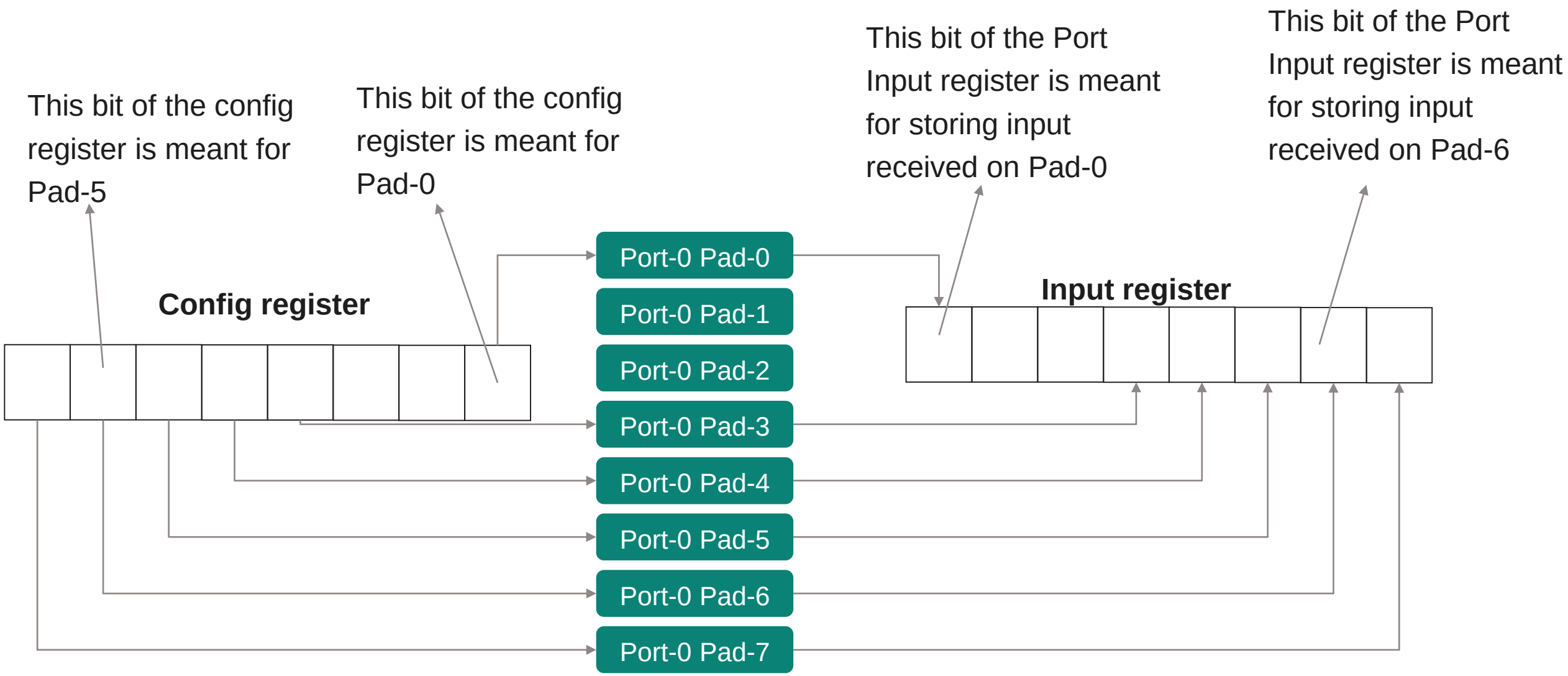
On the previous slide, we said that each pad has a set of user programmable registers.

In reality, providing a set of programmable registers per pad is unwieldy.

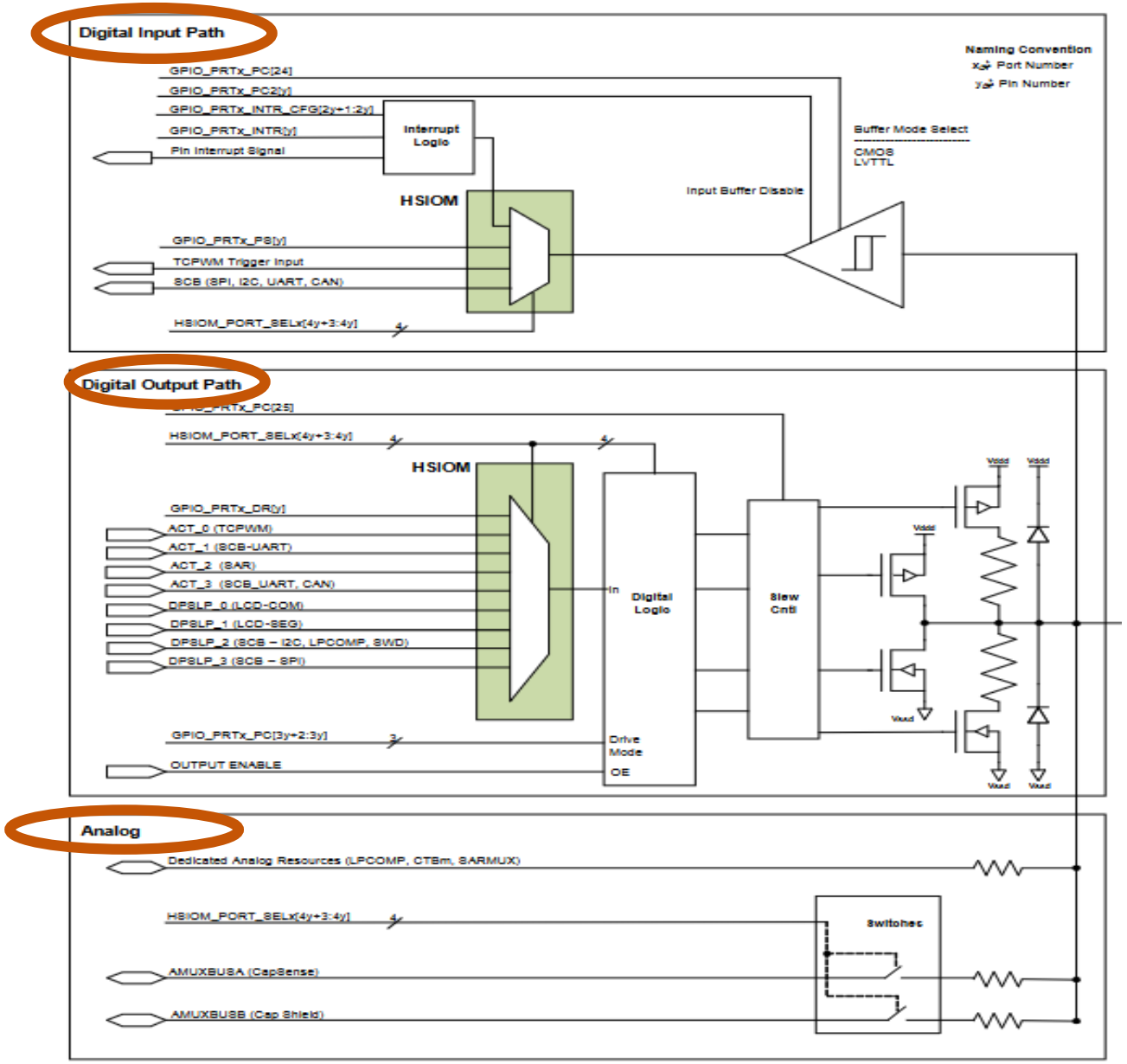
Instead, these register sets are provided for a collection of pads grouped together as a PORT.



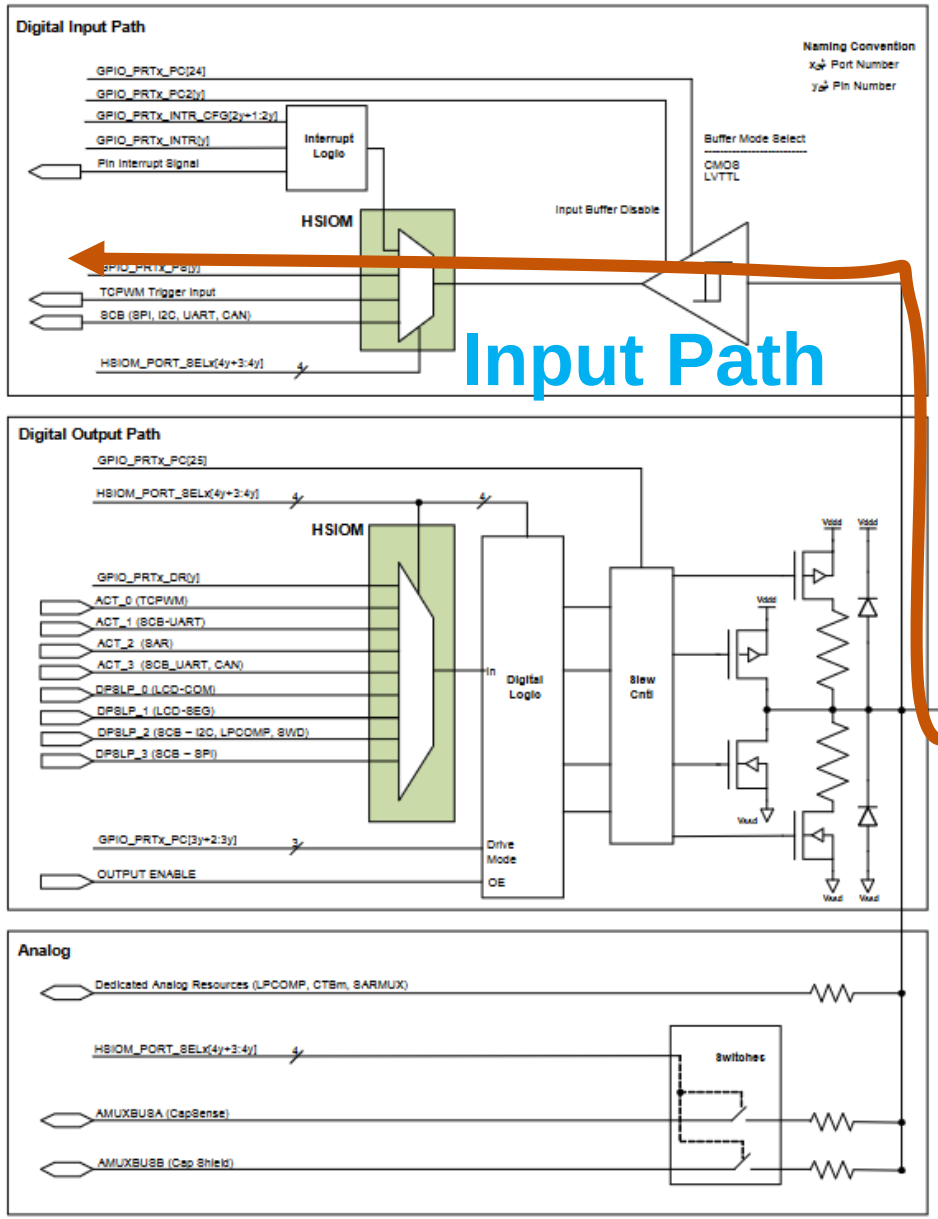
PORT – 2/2

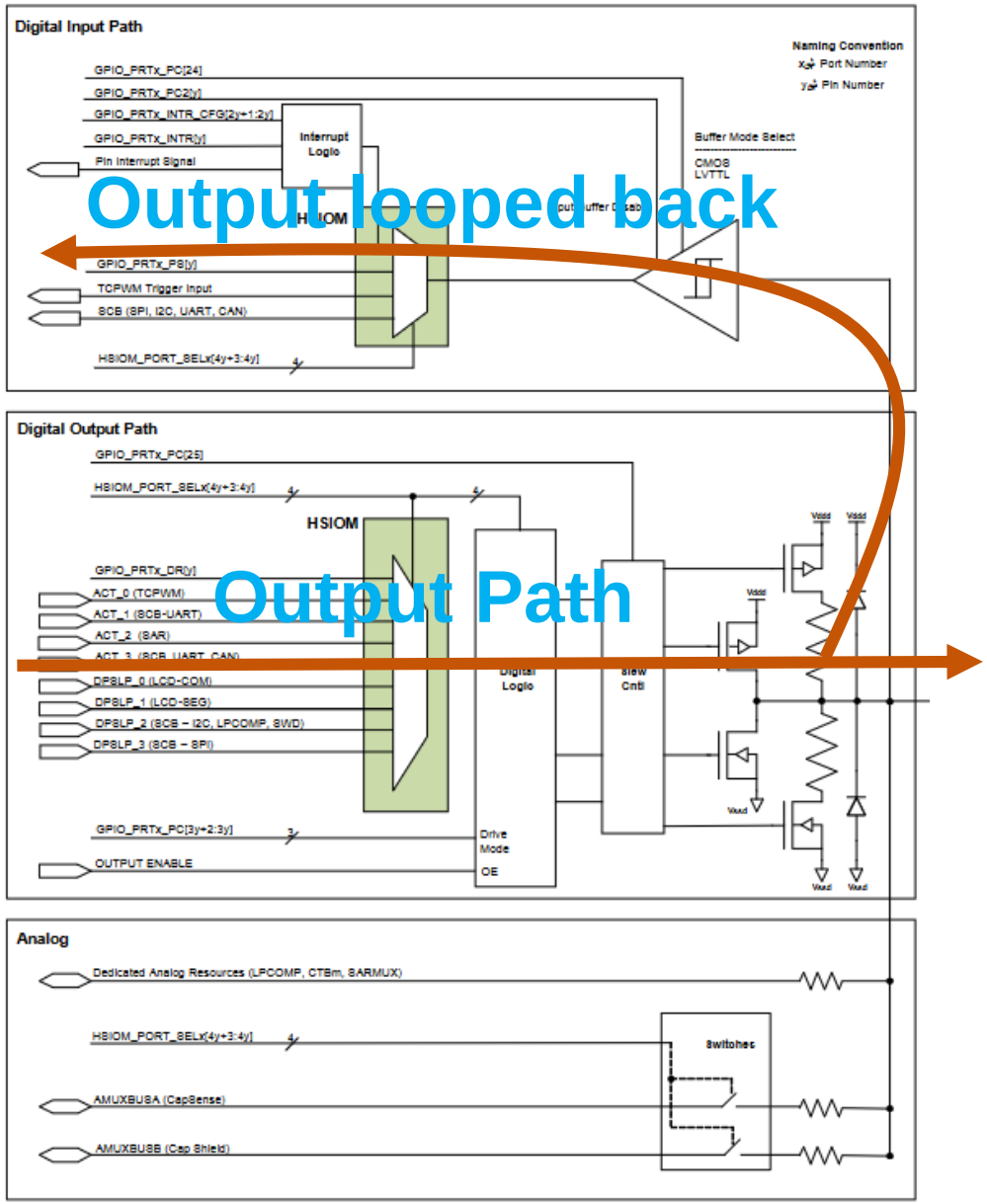


PSoC PORT structure – 1/2



PSoC PORT structure – 2/2





PSoC registers for PORT management (8 PORTS in all)

GPIO_PRT0_DR	0x40040000
GPIO_PRT0_PS	0x40040004
GPIO_PRT0_PC	0x40040008
GPIO_PRT0_INTR_CFG	0x4004000C
GPIO_PRT0_INTR	0x40040010
GPIO_PRT0_PC2	0x40040018
GPIO_PRT0_DR_SET	0x40040040
GPIO_PRT0_DR_CLR	0x40040044
GPIO_PRT0_DR_INV	0x40040048

GPIO_PRT1_DR	0x40040100
GPIO_PRT1_PS	0x40040104
GPIO_PRT1_PC	0x40040108
GPIO_PRT1_INTR_CFG	0x4004010C
GPIO_PRT1_INTR	0x40040110
GPIO_PRT1_PC2	0x40040118
GPIO_PRT1_DR_SET	0x40040140
GPIO_PRT1_DR_CLR	0x40040144
GPIO_PRT1_DR_INV	0x40040148

GPIO_PRT2_DR	0x40040200
GPIO_PRT2_PS	0x40040204
GPIO_PRT2_PC	0x40040208
GPIO_PRT2_INTR_CFG	0x4004020C
GPIO_PRT2_INTR	0x40040210
GPIO_PRT2_PC2	0x40040218
GPIO_PRT2_DR_SET	0x40040240
GPIO_PRT2_DR_CLR	0x40040244
GPIO_PRT2_DR_INV	0x40040248

GPIO_PRT3_DR	0x40040300
GPIO_PRT3_PS	0x40040304
GPIO_PRT3_PC	0x40040308
GPIO_PRT3_INTR_CFG	0x4004030C
GPIO_PRT3_INTR	0x40040310
GPIO_PRT3_PC2	0x40040318
GPIO_PRT3_DR_SET	0x40040340
GPIO_PRT3_DR_CLR	0x40040344
GPIO_PRT3_DR_INV	0x40040348

GPIO_PRT4_DR	0x40040400
GPIO_PRT4_PS	0x40040404
GPIO_PRT4_PC	0x40040408
GPIO_PRT4_INTR_CFG	0x4004040C
GPIO_PRT4_INTR	0x40040410
GPIO_PRT4_PC2	0x40040418
GPIO_PRT4_DR_SET	0x40040440
GPIO_PRT4_DR_CLR	0x40040444
GPIO_PRT4_DR_INV	0x40040448

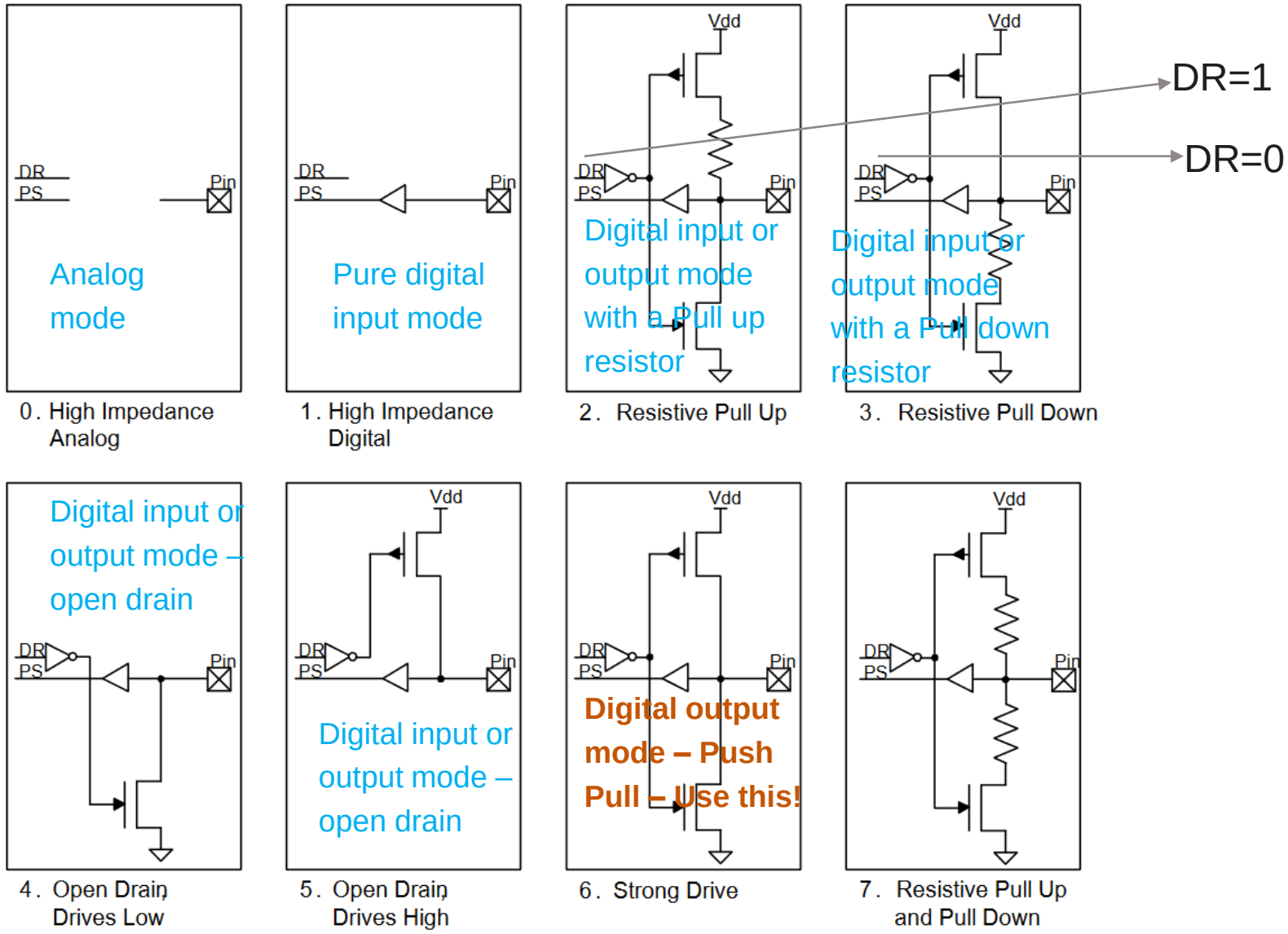
- There are a total of 8 ports (PORT-0 through PORT-7)
- Each PORT houses 8 PADS
- Total: 64 PADS
- There are 10 registers per PORT
- You see 9 of them here
- There is one register HSIOM_PORT_SEL register for MUX selection
- Please refer to user manual for locations of the remaining PORTs (Port-5 through Port-7)

Quick description of PSoC Port registers

PORT Register	Description
HSIOM_PORT_SELx	There are 4 bits for each PAD and can be programmed to select one of the 16 signal options. **Default option: GPIO**
GPIO_PRT_DR	This is the output data register. One bit per PAD. Write ‘0’ or ‘1’ in the relevant bit position and that value will appear as Pin output
GPIO_PRT_PS	This is the input data register. One bit per PAD. Read the relevant bit position to determine the value (0 or 1) of the digital input appearing at pin.
GPIO_PRT_PC	3 bits per PAD. So a total of 8 pad options to choose from. See next slide.
GPIO_PRT_PC2	1 bit per PAD. Set relevant bit to 1 when that PAD is meant to serve analog inputs.
GPIO_PRT_DR_SET	1 bit per PAD. Write a ‘1’ in relevant bit position to drive a ‘1’ on to the pin.
GPIO_PRT_DR_CLR	1 bit per PAD. Write a ‘1’ in relevant bit position to drive a ‘0’ on to the pin.
GPIO_PRT_INTR_CFG	2 bits per PAD. You can program these bits to generate an interrupt event.

GPIO_PRT_PC – 1 of 2

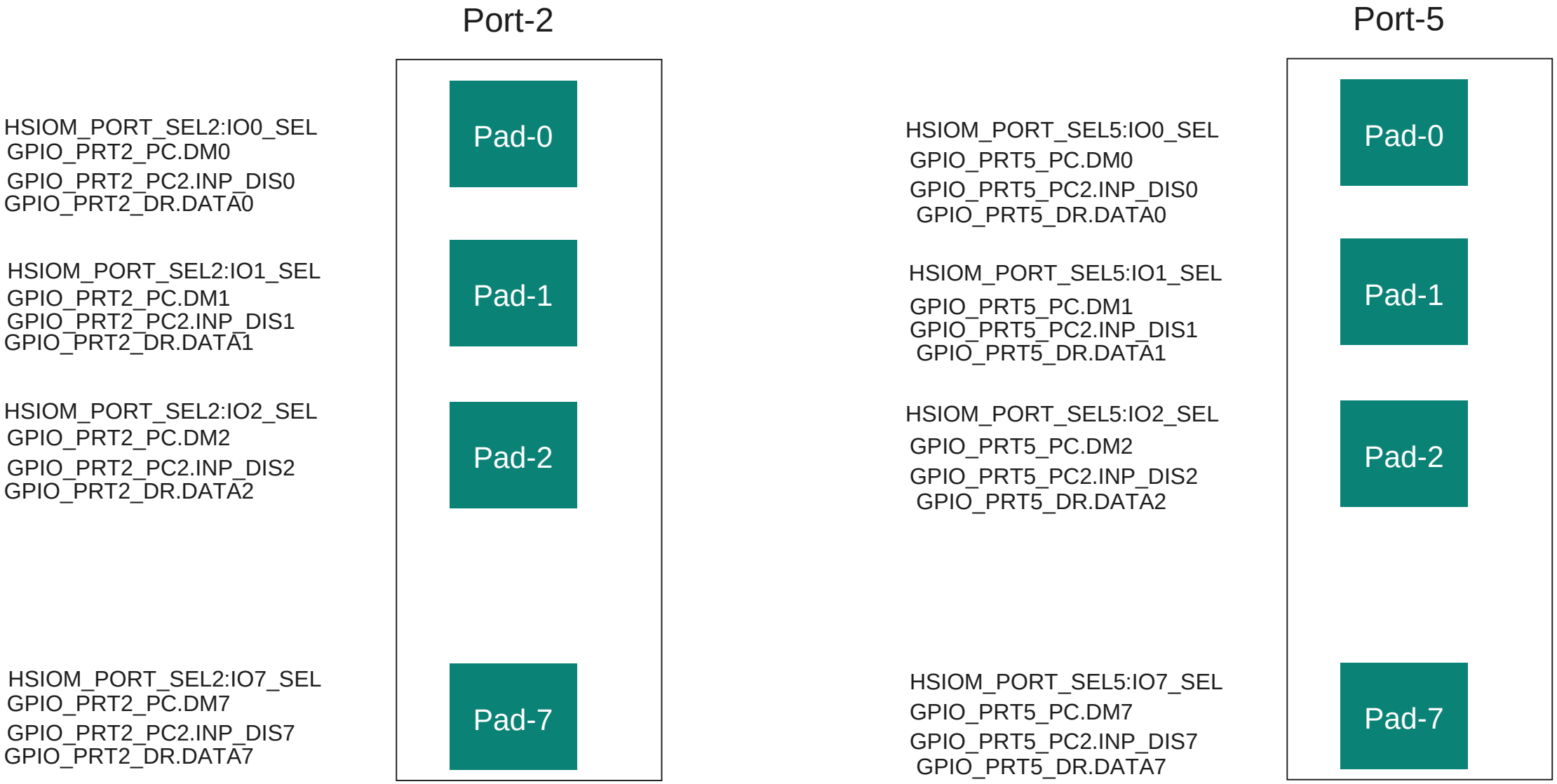
PAD Drive modes



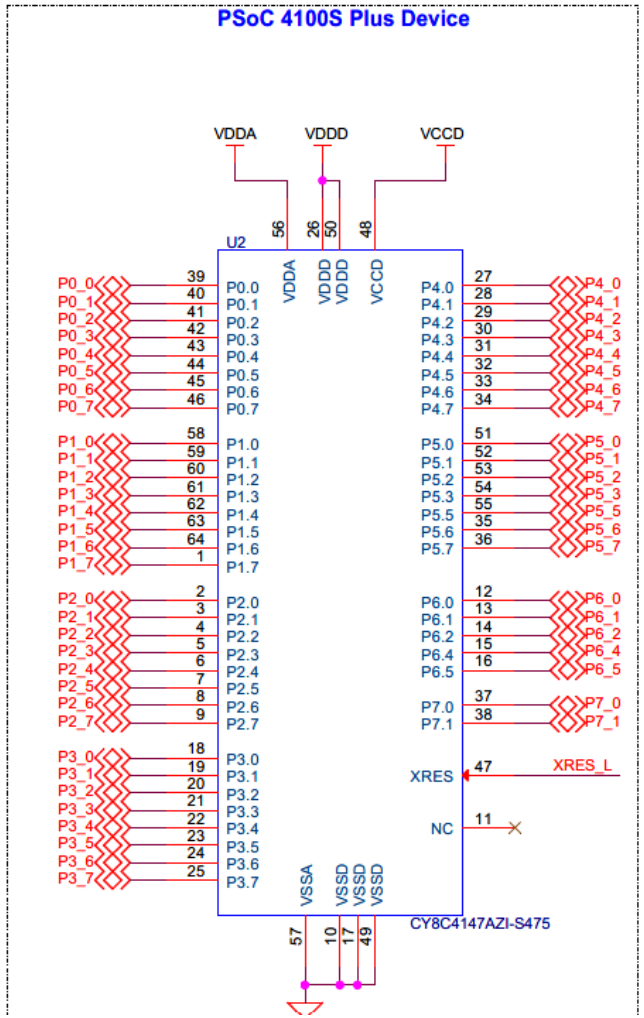
GPIO_PRT_PC – 2 of 2

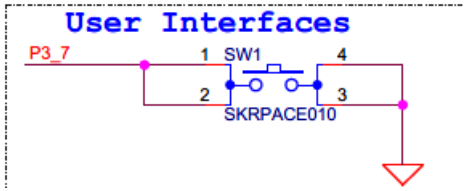
DM	Configuration	Comment
0	Analog Input	Digital output and Digital input stages are disabled. A signal entering the pad is diverted to Analog Input section
1	Digital Input	Digital output and Analog input stages are disabled. A signal entering the pad is diverted to Digital Input section.
2	Digital IO	Digital input and output sections are enabled with the Resistive Pull-Up option.
3	Digital IO	Digital input and output sections are enabled with the Resistive Pull-Down option.
4	Digital IO	Digital input and output sections are enabled in Open-Drain mode. If a 0 is written, that 0 will appear on the line. If a 1 is written, the line floats due to open drain.
5	Digital IO	Digital input and output sections are enabled in Open-Drain mode. If a 1 is written, that 1 will appear on the line. If a 0 is written, the line floats due to open drain.
6	Digital Output	Digital output section is enabled with Push-Pull transistor drive configuration.
7	Homework	Homework

Pads and associated registers to program



Example-1 - 1/5



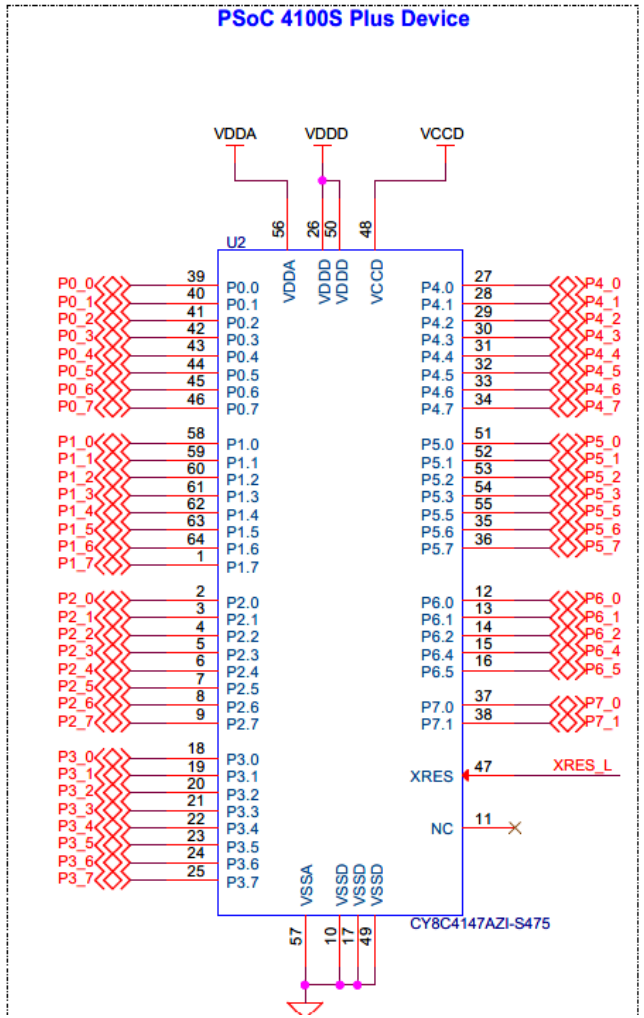


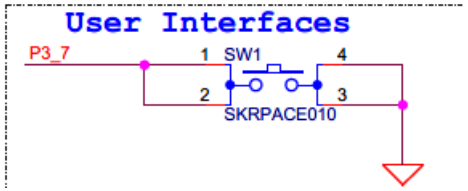
- One of the terminals of the switch SW1 is connected to ground
- The other terminal is connected to Port 3 Pad 7 of the microcontroller
- The switch signal is meant to be evaluated by software and based on the switch position, certain actions are taken by software
- When the switch is pushed down, Logic-0 or Ground is delivered to P3_7
- When the switch is not pressed, we expect Logic-1 to be delivered to P3_7

- Question: How should Pad7 of Port3 be configured?
- Ans: Definitely as an input pad

- Question: But which of the 3 input modes?
- Answer: For this, we must look at the circuit (Contd)

Example-1 - 2/5





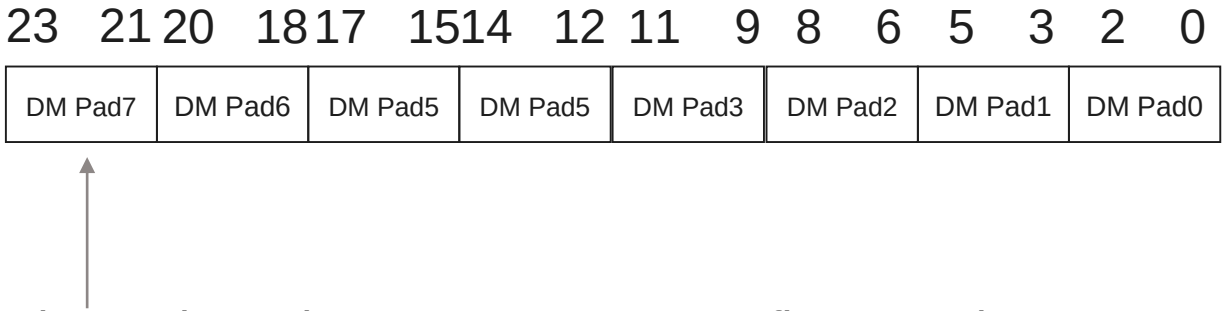
- Question: How should Pad7 of Port3 be configured?
- Ans: Definitely as an input pad

- Question: But which of the 3 input modes?
- Answer: For this, we must look at the circuit

- P3_7 must only receive a 0 when the switch is pressed
- Otherwise at all other times, it must receive a 1
- For this, we require a Pull-Up resistor
  - We examine the schematic and do NOT find an external Pull-up resistor on the line
  - So we must use an internal pull up resistor mechanism
  - The correct choice here is therefore DM=0x2 (Resistive Pull up)
  - This alone is NOT sufficient
  - For Vdd to be connected to the internal pull-up resistor, the upper transistor must be turned ON
  - For this, program, the Data Register (DR) bit of the pad to 1
  - The NOT gate will invert DR bit and turn ON the transistor

Example-1 – 3/5

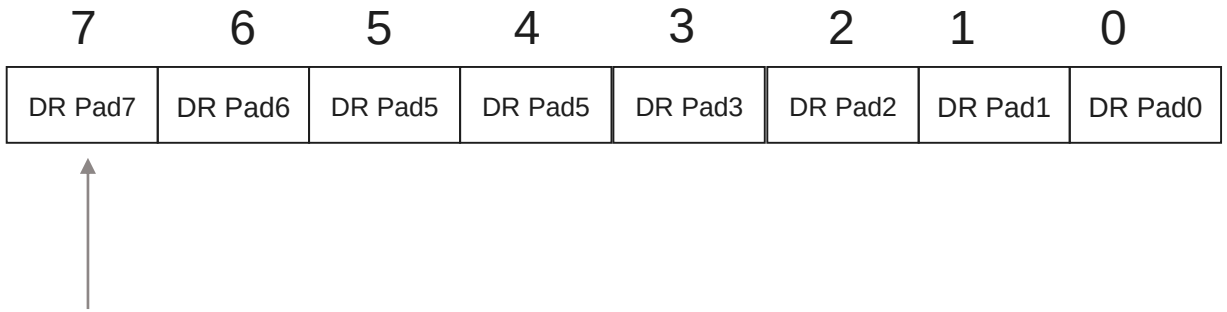
GPIO_PRT3_PC



Bits 23 through 21 are meant to configure Pad-7

Since we have chosen DM=0x2, we write 010 into these bits

GPIO_PRT3_DR

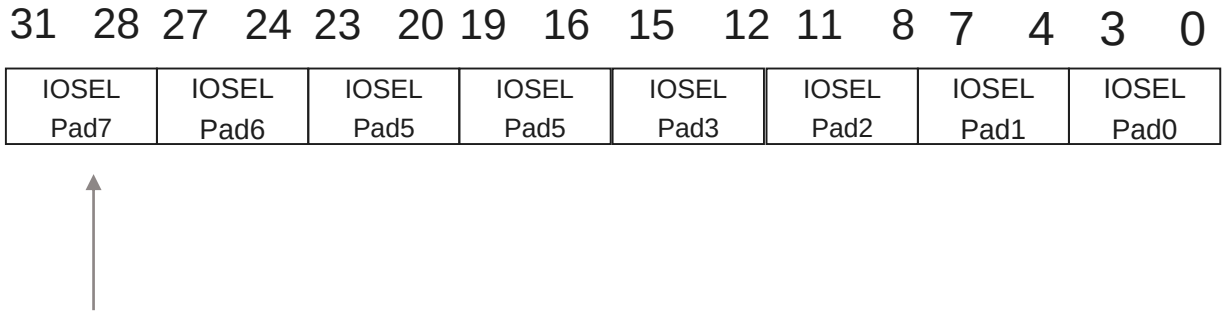


Bit-7 corresponds to Pad-7

Since we have chosen DM=0x2, we write ‘1’ here to turn ON the upper transistor [See Slide-24 to understand the significance of this]

Example-1 – 4/5

HSIOM_PORT_SEL3



- This is a 4 bit field.
- There are 16 options to choose from
- GPIO is Option-0
- Therefore we write “0000” into these bit positions
- **Tip:** GPIO is the default selection. If your choice is GPIO, this update may not be required at all.

Example-1 – 5/5

- Final programming sequence
  - Program Bits 31:28 of HSIO_PORT_SEL3 to a value of “0000” to configure the pad as GPIO
  - Program Bits 23:21 of GPIO_PRT3_PC to a value of “010” to configure drive mode as Input with Pull up
  - Program Bit 7 of GPIO_PRT3_DR to a value of ‘1’ to turn on the pass transistor that connects Vcc to the Pull up resistor

- After the programming sequence depicted above, Pad-7 of Port-3 is successfully configured as Input with Pull-up resistor

- You may now determine the input value at the pad input by reading the data input register GPIO_PRT3_PS
- Bit-7 of this GPIO_PRT3_PS register corresponds to Pad-7. So that bit must be read.


Infineon