Heaven's Light Is Our Guide

Rajshahi University of Engineering & Technology
Department of Computer Science & Engineering

CSE 3105
Computer Interfacing & Embedded System

General Purpose I/O (GPIO)

Md. Nasif Osman Khansur
Lecturer
Dept. of CSE, RUET

References

1. Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C 3e by Dr. Yifeng Zhu **[Chapter 14]**

Introduction to General Purpose (GPIO)

- A processor pin that can be configured by software at runtime to perform various functions is called a **general-purpose input/output (GPIO) pin**.
- Software can program a GPIO pin as one of the following four different functions:

1. **Digital input** that detects whether an external voltage signal is higher or lower than a predetermined threshold.
2. **Digital output** that controls the voltage on the pin.
3. **Analog functions** that perform digital-to-analog or analog-to-digital conversion.
4. Other complex functions such as PWM output, LCD driver, timer-based input capture, external interrupt, and interface of USART, SPI, I2C and USB communication. We call the last category of functions as **alternate functions (AF)**. The software can dynamically change the function of a GPIO pin at runtime.

Introduction to General Purpose (GPIO)

- A GPIO port consists of a group of GPIO pins, typically 8 or 16, which share the same data and control registers.
  1. When a GPIO pin ***i*** is set as a **digital input**, the binary data read from this pin of this GPIO group is saved at bit ***i*** in the input data register (IDR). Each bit in IDR holds the digital input of the corresponding pin.
  2. When a GPIO pin ***i*** is configured as a **digital output**, bit ***i*** in the output data register (ODR) holds the output of this pin. Therefore, when changing the output of a GPIO pin, the programmer should only alter the value of the corresponding bit of ODR, without affecting the other bits in ODR.
  3. All GPIO pins in a GPIO port can be configured as input or output independently.

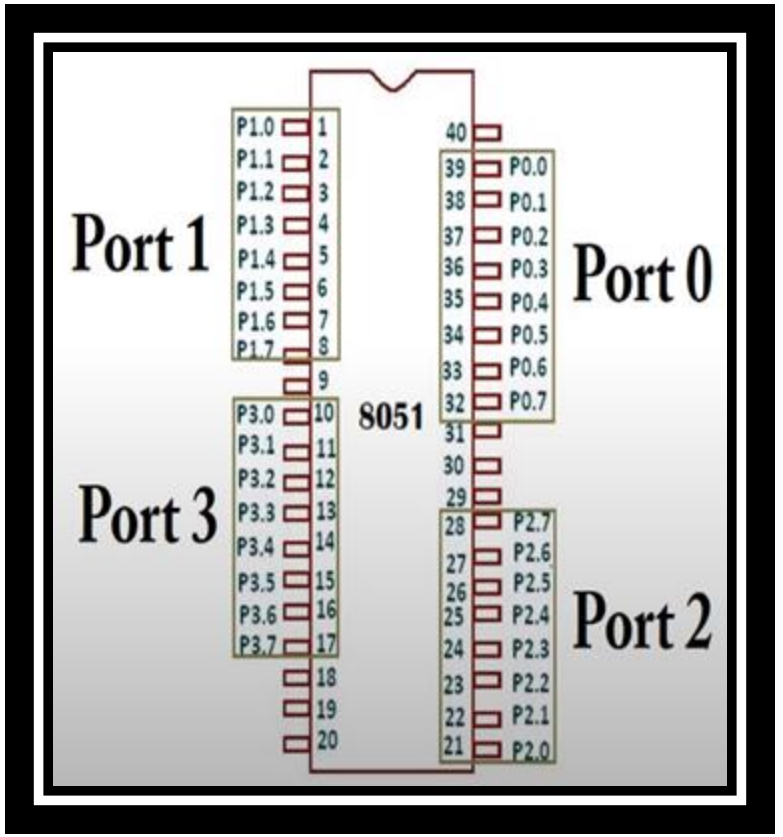


Fig. 8051 Microcontroller

Introduction to General Purpose (GPIO)

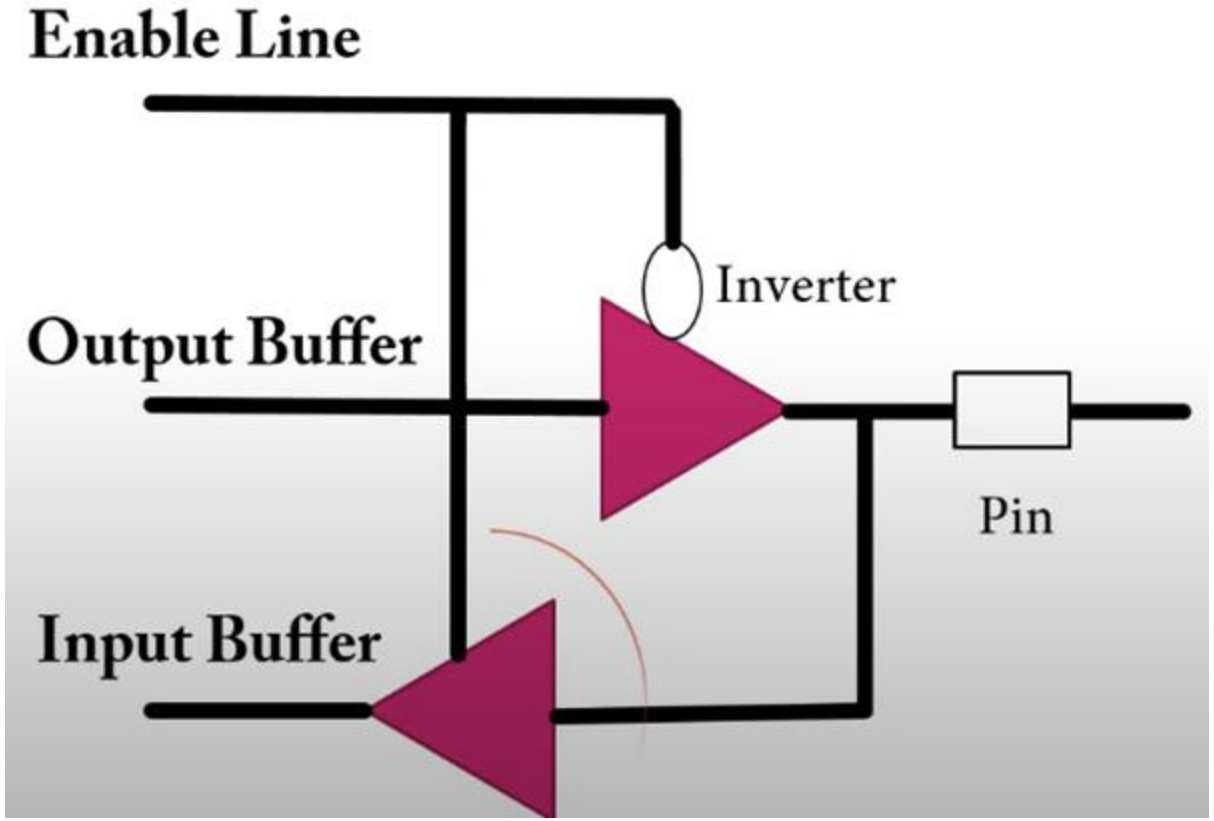
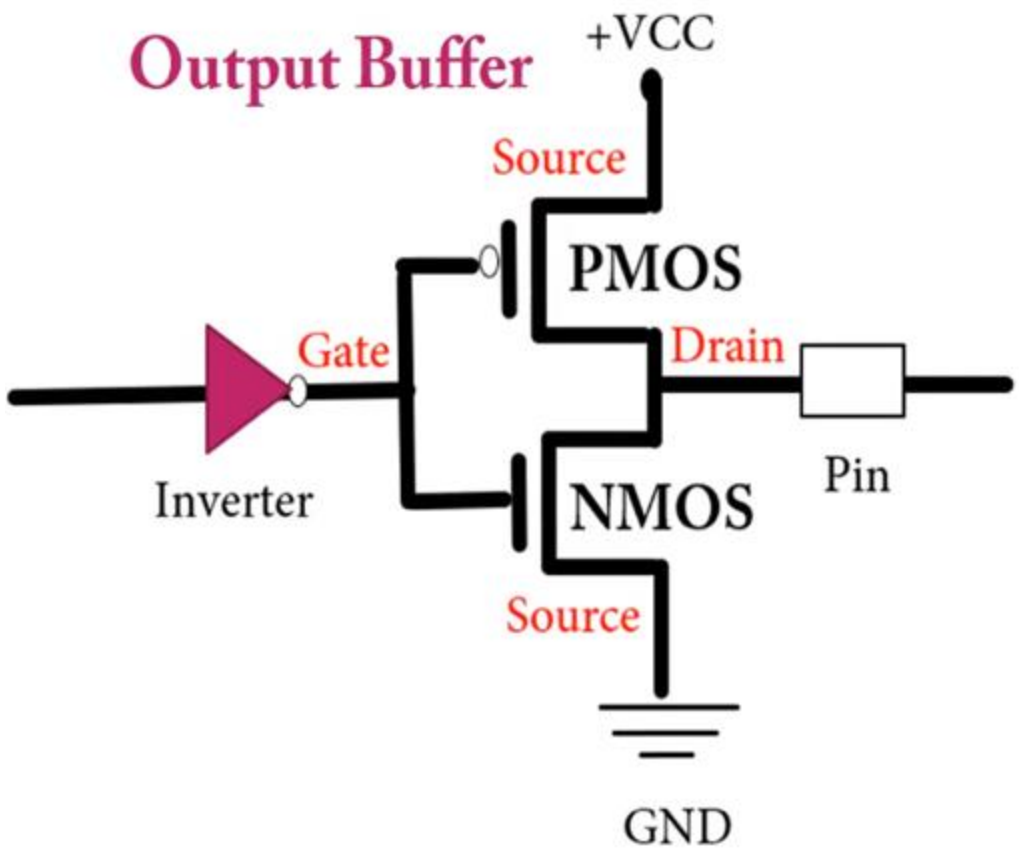
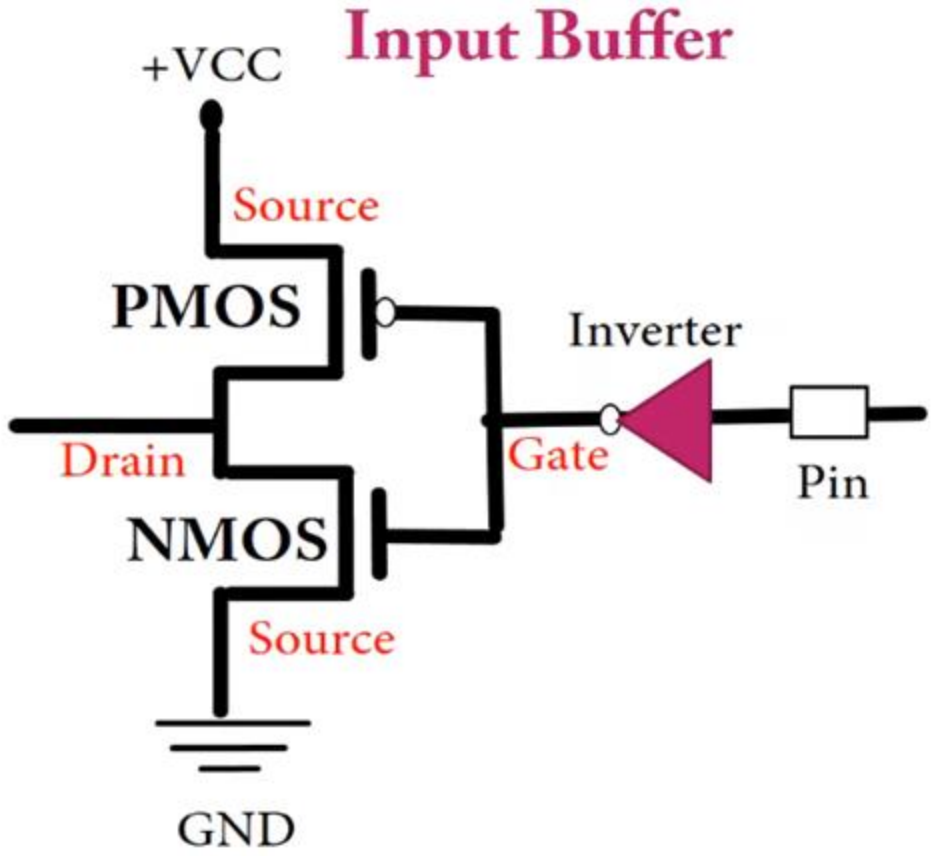


Fig. A GPIO Pin

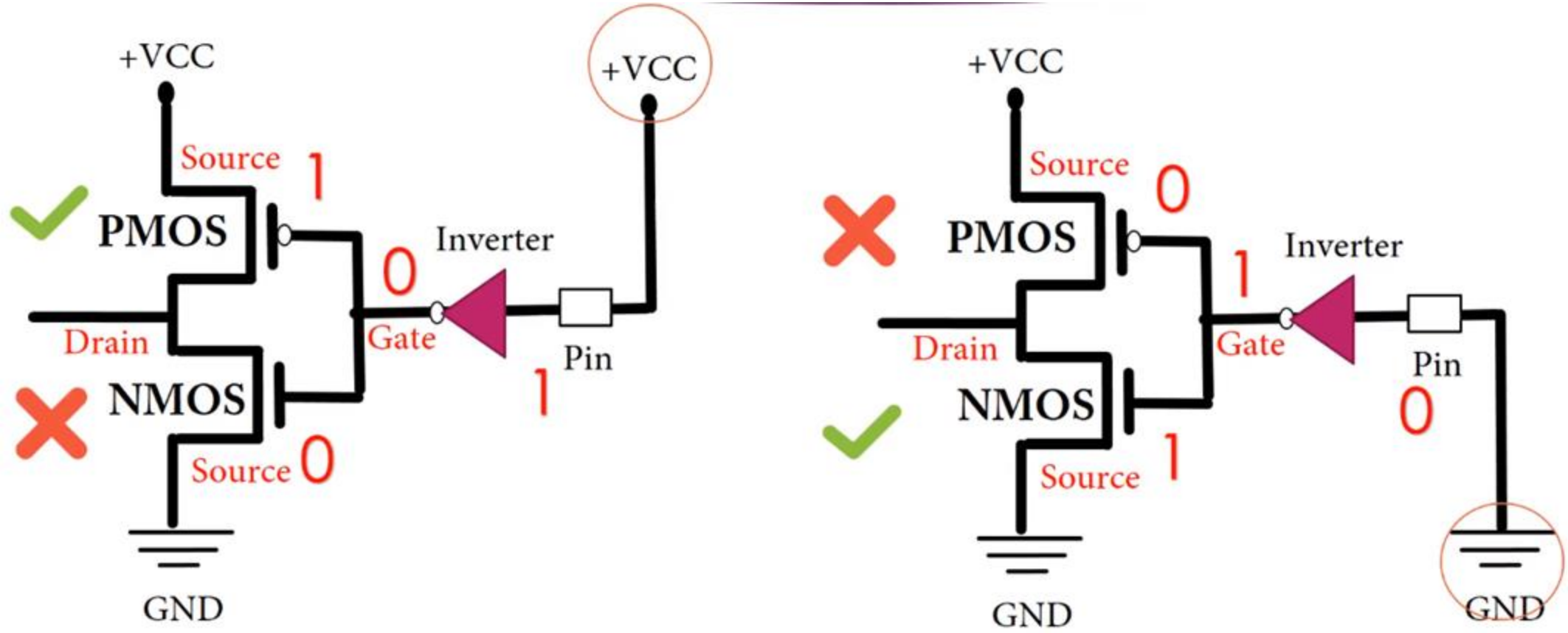
Introduction to General Purpose (GPIO)





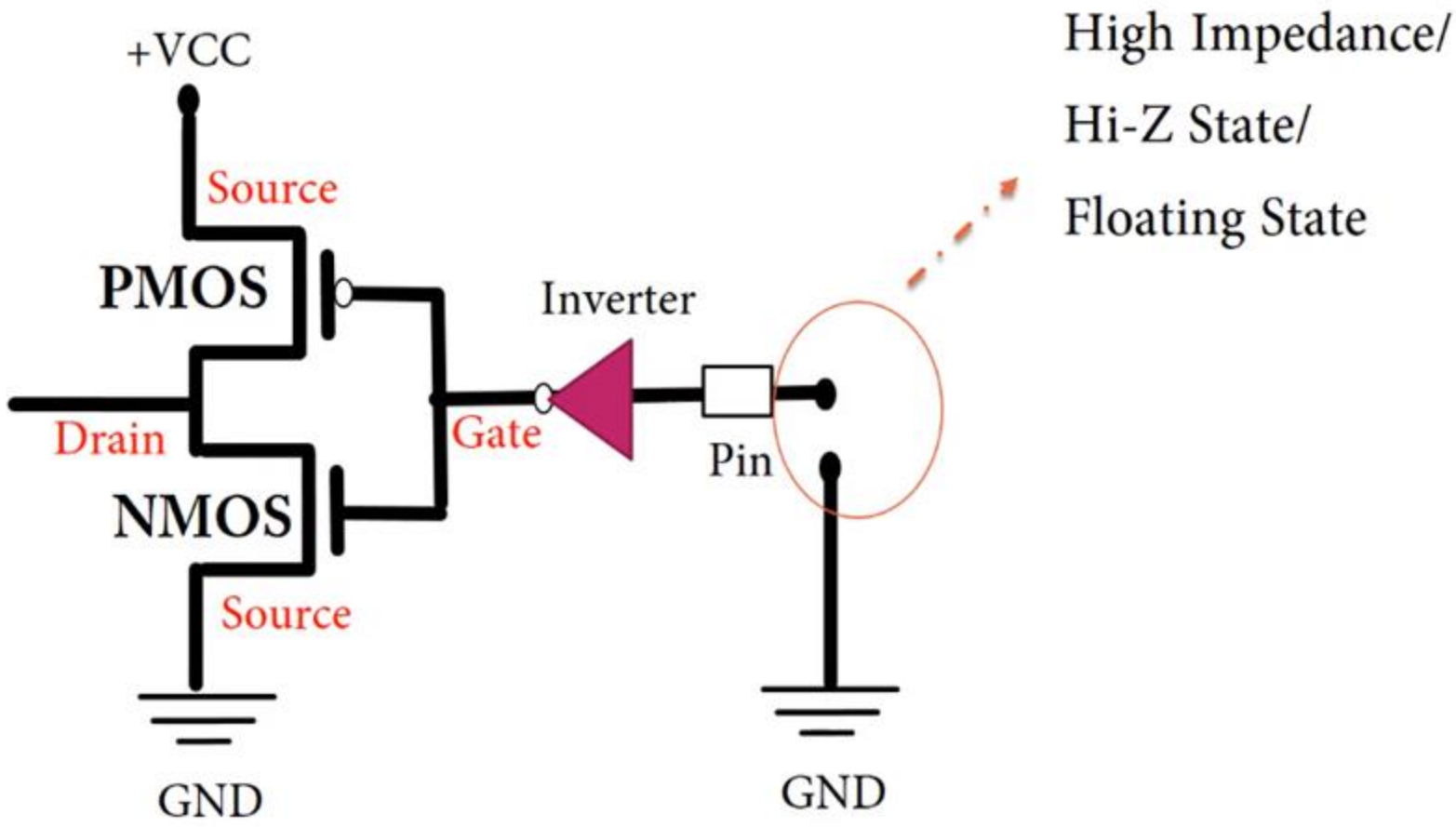
Introduction to General Purpose (GPIO)

When a GPIO pin is used as digital input, the pin has three states: **high voltage, low voltage, or high impedance (also called floating or tri-stated)**.



Introduction to General Purpose (GPIO)

When a GPIO pin is used as digital input, the pin has three states: **high voltage, low voltage, or high impedance (also called floating or tri-stated)**.



GPIO Input Modes: Pull Up and Pull Down

- **Pull Up and Pull Down:** Pull-up and pull-down are used to ensure the input pin has a valid high (logic 1) or a valid low (logic 0) when the external circuit does not drive the pin.
  1. **Pull Up:** When software configures a pin as pull-up, the pin is internally connected to the power 1) is supply via a resistor, as shown in Figure 14-1. The pin always read as high (logic 1) unless the external circuit drives this pin low.
  2. **Pull Down:** When a pin is configured as pull-down, the pin is then internally connected to the ground via a resistor, as shown in Figure 14-2. The pin is always read as low (logic 0) unless the external circuit drives this pin high.
- When a pin is neither pulled up nor pulled down internally, then pin has high impedance, and analog signal on the GPIO pin cannot reliably represent a logic value.

GPIO Input Modes: Pull Up and Pull Down

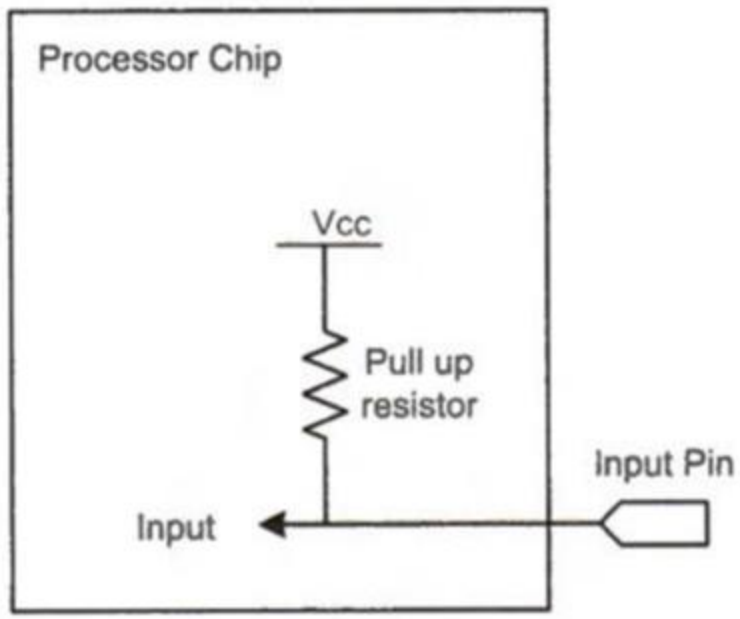


Figure 14-1. The GPIO pin is pulled up internally.

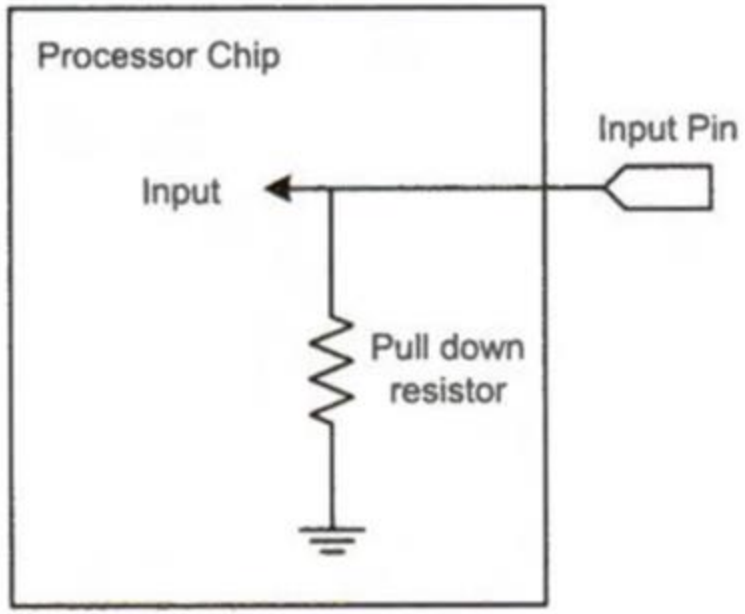


Figure 14-2. The GPIO pin is pulled down internally.

Weak Pull vs Strong Pull

- When an external circuit connected to a GPIO pin has a fair amount of capacitance, the process of pulling the pin voltage to the level of logic high or logic low takes a long time because the impedance of the pull-up and pull-down resistors is too large. We call pulling via large resistors **weak pull-up** or **weak pull-down**.

- To change the pin voltage rapidly, a GPIO pin can be externally pulled up or down via a smaller resistor (several KΩ). Pulling via small resistors is often called **strong pull-up** or **strong pull-down**.

GPIO Input: Schmitt Trigger

- Each GPIO input module usually includes a Schmitt trigger. A Schmitt trigger uses a voltage comparator to convert a noisy or slow signal edge into a clean edge with instantaneous transition.

- In real systems, an input signal from external devices usually cannot change instantly. Such input signal tends to have a low slew rate because of inherent parasitic capacitance, resistance, or induction in the input data path. A processor chip usually has built-in Schmitt triggers to increase slew rate and enhance noise immunity for external input signals.

Schmitt Trigger vs Simple Comparator

- Figure 14-5 compares the output voltage V_{out} of Schmitt trigger and a simple comparator when the input signal varies irregularly.
  - ❑ Compared with a simple comparator, Schmitt trigger provides better noise rejection. The threshold of Schmitt trigger is larger than that of a simple comparator for switching high, and lower for switching low.
  - ❑ If the input signal fluctuates slightly, output of Schmitt trigger does not change. For this reason, the Schmitt trigger is immune to undesired noise.

Schmitt Trigger vs Simple Comparator

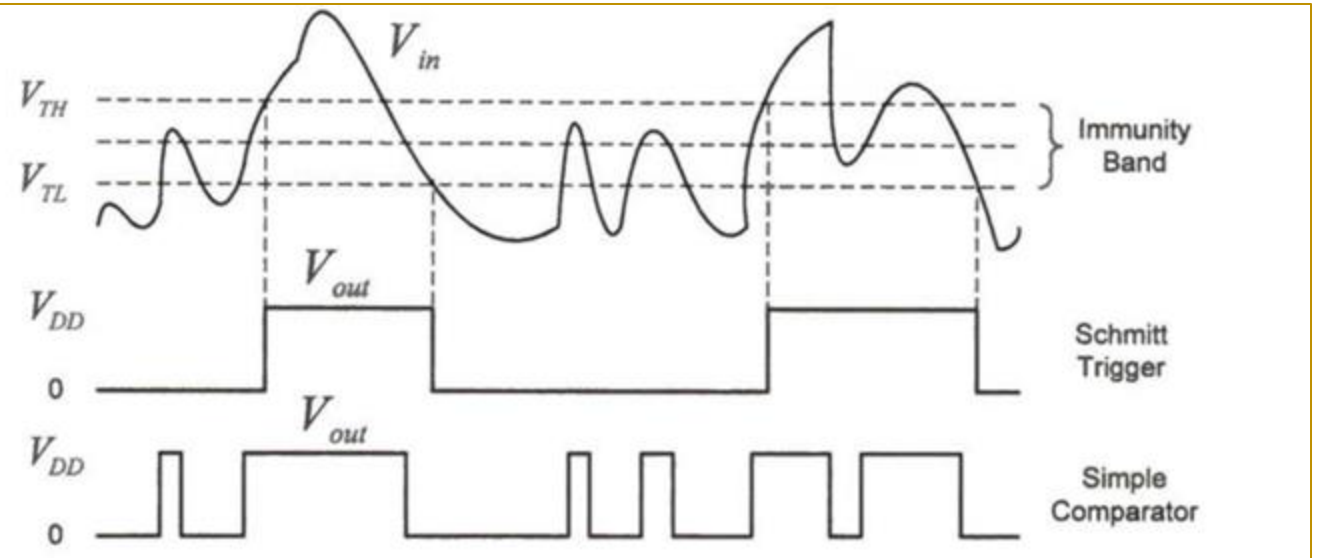


Figure 14-5. Comparing the voltage output of Schmitt trigger with a simple comparator. Schmitt trigger converts an irregular-shaped signal V_{in} into a square wave V_{out} based on two switching thresholds. The simple comparator uses a single threshold (the dotted line between V_{TL} and V_{TH}) to generate the output.

GPIO Output Modes: Push-Pull and Open-Drain

- Software can configure a GPIO output pin as either push-pull or open-drain.
  - Push-pull mode allows the pin to supply and absorb current.
  - However, a GPIO pin in open-drain (also called collector) mode can only absorb current.

GPIO Output Modes: Push-Pull

- A push-pull output consists of a pair of complementary transistors, as shown in figure 14-6. Only one of them is turned on at any time.
- One transistor pushes the output on a positive half-cycle and the other pulls on a negative half cycle.

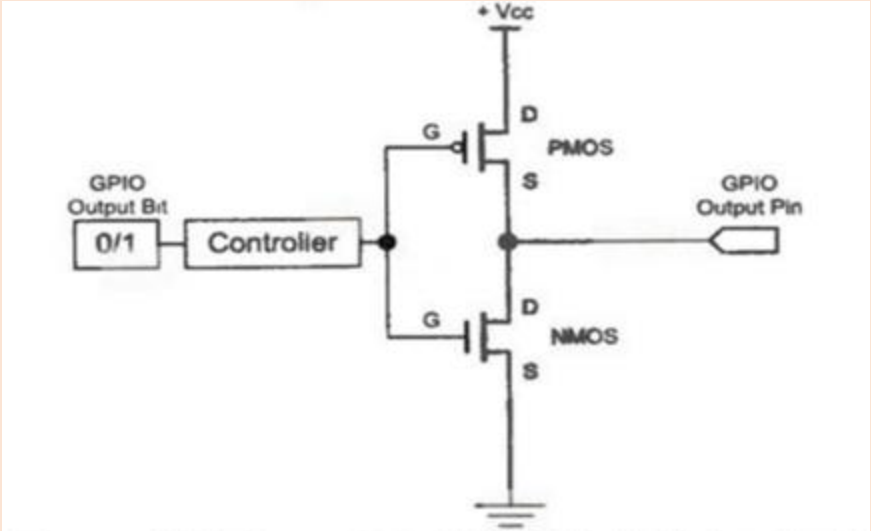


Figure 14-6. A push-pull GPIO digital output

GPIO Push-Pull Output

- When logic 0 is outputted, the transistor connected to the ground is turned on to sink an electric current from the external circuit, as shown in Figure 14-7.
- When the pin outputs logic 1, the transistor connected to the power supply is turned on, and it provides an electric current to the external circuit connected to the output pin, as shown in Figure 14-8.

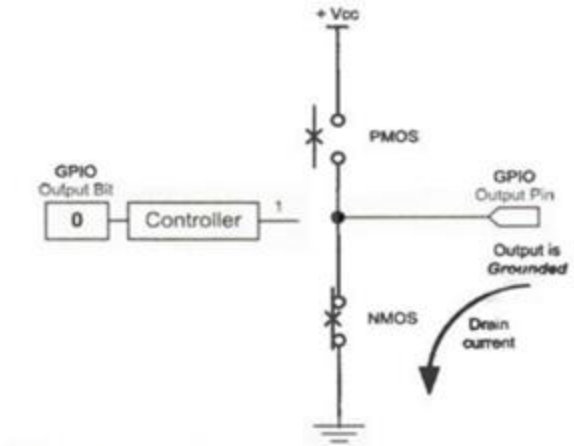


Figure 14-7. If the digital output is 0, then the GPIO output pin is pulled down to the ground in a push-pull setting.

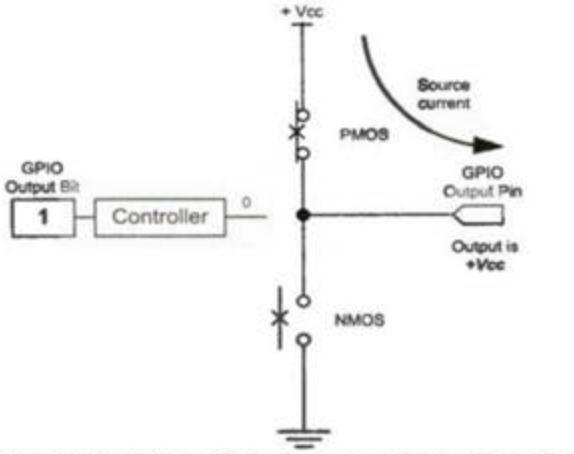


Figure 14-8. If the digital output is 1, then the GPIO output pin is pulled up to the Vcc in a push-pull setting.

GPIO Open-Drain Output

- An open-drain output consists of a pair of the same type of CMOS or transistors, as shown in Figure 14-9.
- An open-drain output has only two states: low voltage (logic 0), and high impedance (logic 1). **It often has an external pull-up resistor.**

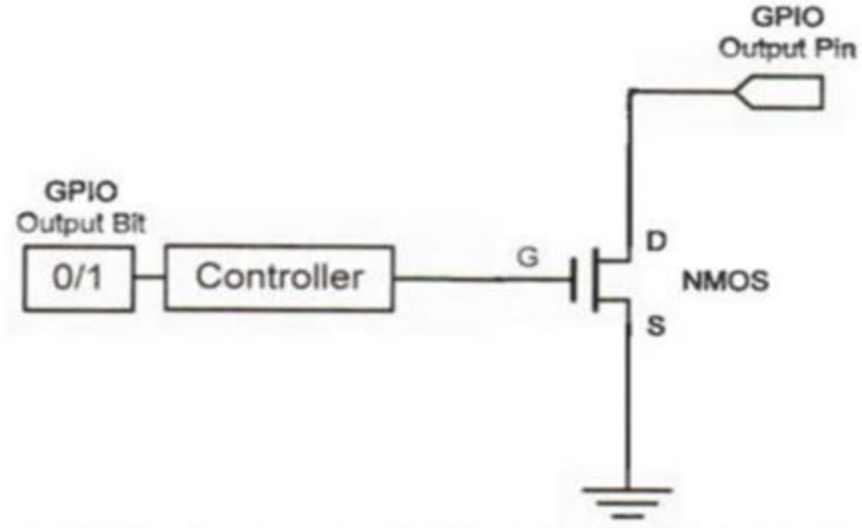


Figure 14-9. An open-drain GPIO digital output

GPIO Open-Drain Output

- When software outputs a logic 0, open-drain circuit can sink an electric current from the external load connected to the GPIO pin.
- When software outputs a logic 1, it cannot supply any electric current to the external load because the output pin is floating, connected to neither the power supply nor the ground.

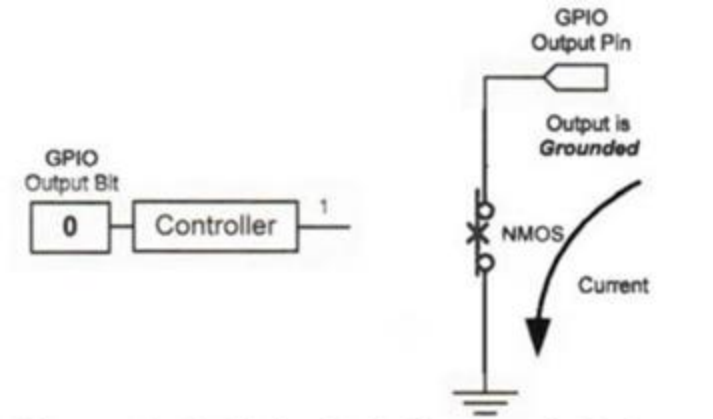


Figure 14-10. If the digital output is 0, then the output pin is pushed to the ground in an open-drain setting (the scenario of *drain*).

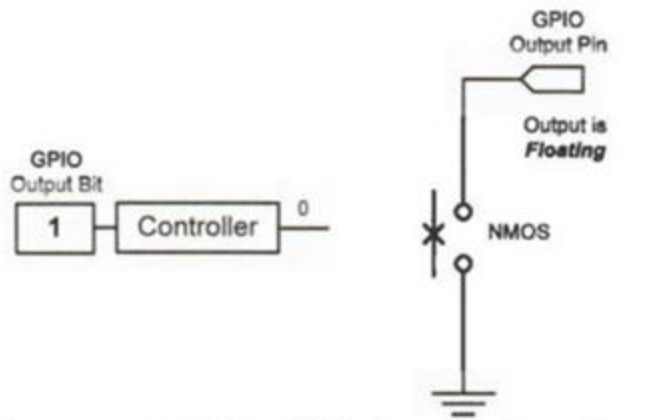


Figure 14-11. If the digital output is 1, then the output pin is floating in an open-drain setting (the scenario of *open*).

Usage of GPIO Open-Drain Outputs

- One important usage of open-drain outputs is to connect directly several outputs together and implement wired logic AND (active high) or OR (active low) circuit in an easy way. If multiple open-drain output pins are connected and are pulled up via a shared resistor, any output pin can drive the output voltage low. The pin voltage is high if and only if all pins output a high voltage level.
  1. If a high voltage level represents logic state 1 (active high), it implements a wired-AND function. The final output is 1 (high) only if all outputs of connected pins are 1 (high).
  2. If a low voltage level represents logic state 0 (active low), it implements a wired-OR function. The final output is 1 (low) if the output of any pins is 1 (low).

Usage of GPIO Open-Drain Outputs

Figure 14-12 shows the implementation of wired-AND by using open drain and external pull-up when active high logic is used. The output C is determined by the following table.

Inputs				Output
Logic A	Logic B	Circuit A	Circuit B	
0	0	Drain	Drain	0
0	1	Drain	Open	0
1	0	Open	Drain	0
1	1	Open	Open	1

Usage of GPIO Open-Drain Outputs

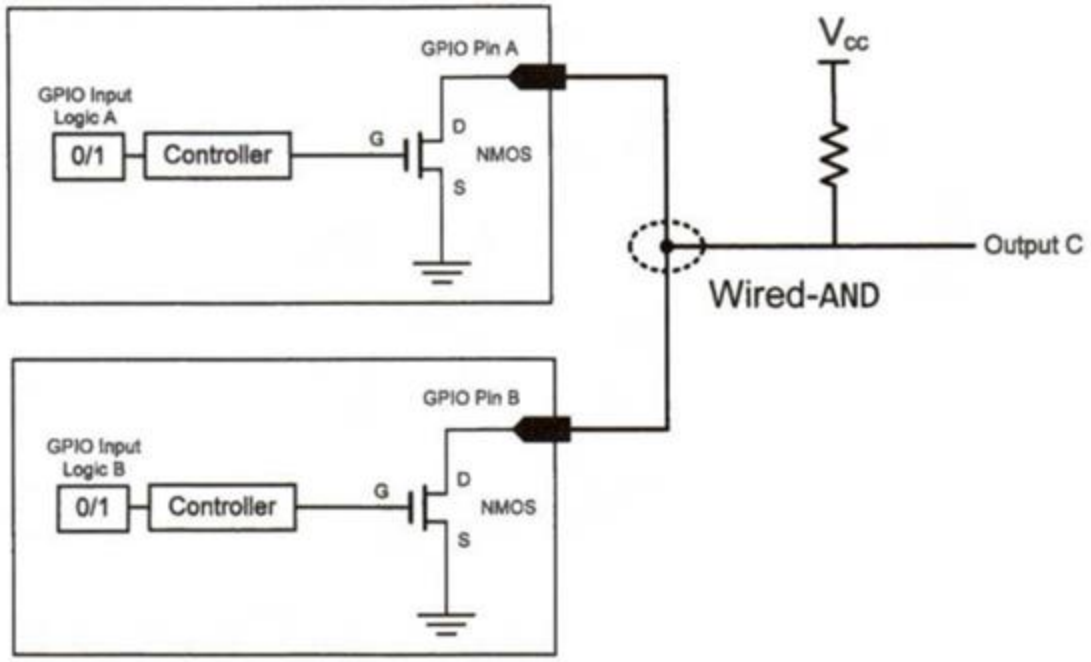


Figure 14-12. Implementation of Wired AND by using open drain and external pull up

Open-Drain vs Push Pull

- Compared to open-drain, push-pull mode has the advantage of faster speed, because it can change the pin voltage faster if the external circuit has some capacitance.

- Another advantage is that it can supply current and simplify the circuit. For example, a push-pull output can directly control an external LED while an open-drain output cannot light up an LED without external voltage source.

GPIO Output Speed: Slew Rate

- The slew rate of a GPIO pin is the speed of change of its output voltage per unit of time as defined as follows

$$Slew\ Rate = \frac{\Delta V}{\Delta t}$$

- If the logic output of a GPIO pin changes from 0 to 1 and accordingly the voltage output of the pin rises from 0V to 3V in 3µS, then the slew rate is 1V per µS

GPIO Output Speed: Slew Rate

- Figure 14-13 shows an example of ΔV and Δt when the output voltage increases from low to high. The slew rate definition applies to both the rising edge and the falling edge of a voltage output.

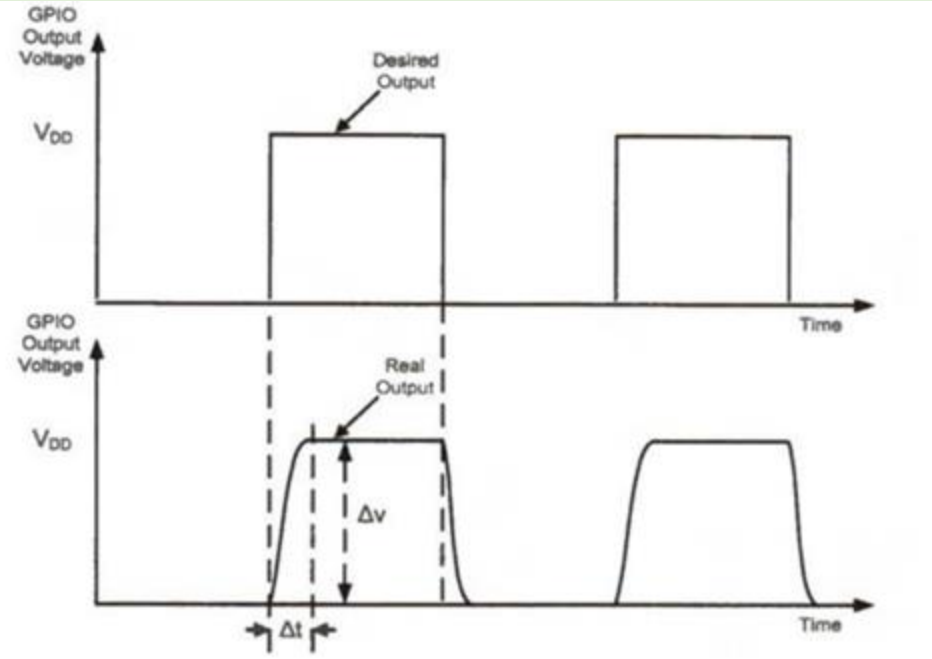


Figure 14-13. Comparing a desired square wave voltage output with the real GPIO output

High Slew Rate vs Slow Slew Rate

- The higher the slew rate, the shorter time the output voltage takes to rise or fall to desired values. Therefore, a higher slew rate allows faster speed at which the processor can toggle the logic level of a GPIO pin.
- However, a large slew rate often causes high electromagnetic interference (EMI), also called radio frequency interference (RFI) to neighbor electronic circuits. A fast rising and falling signal has large-amplitude and high-frequency harmonics, which can transfer to a victim circuit via radiation, conduction, or induction, and may cause malfunctions. A slower valid slew rate is often preferred to minimize EMI disturbance.
- The slew rate of the GPIO circuit is programmable by setting the GPIO output speed. For example, the digital output speed of a GPIO pin can be low speed (400kHz), medium (2 MHz), fast speed (10 MHz), or high speed (40 MHz) in the STM32L processors.

Thank You!