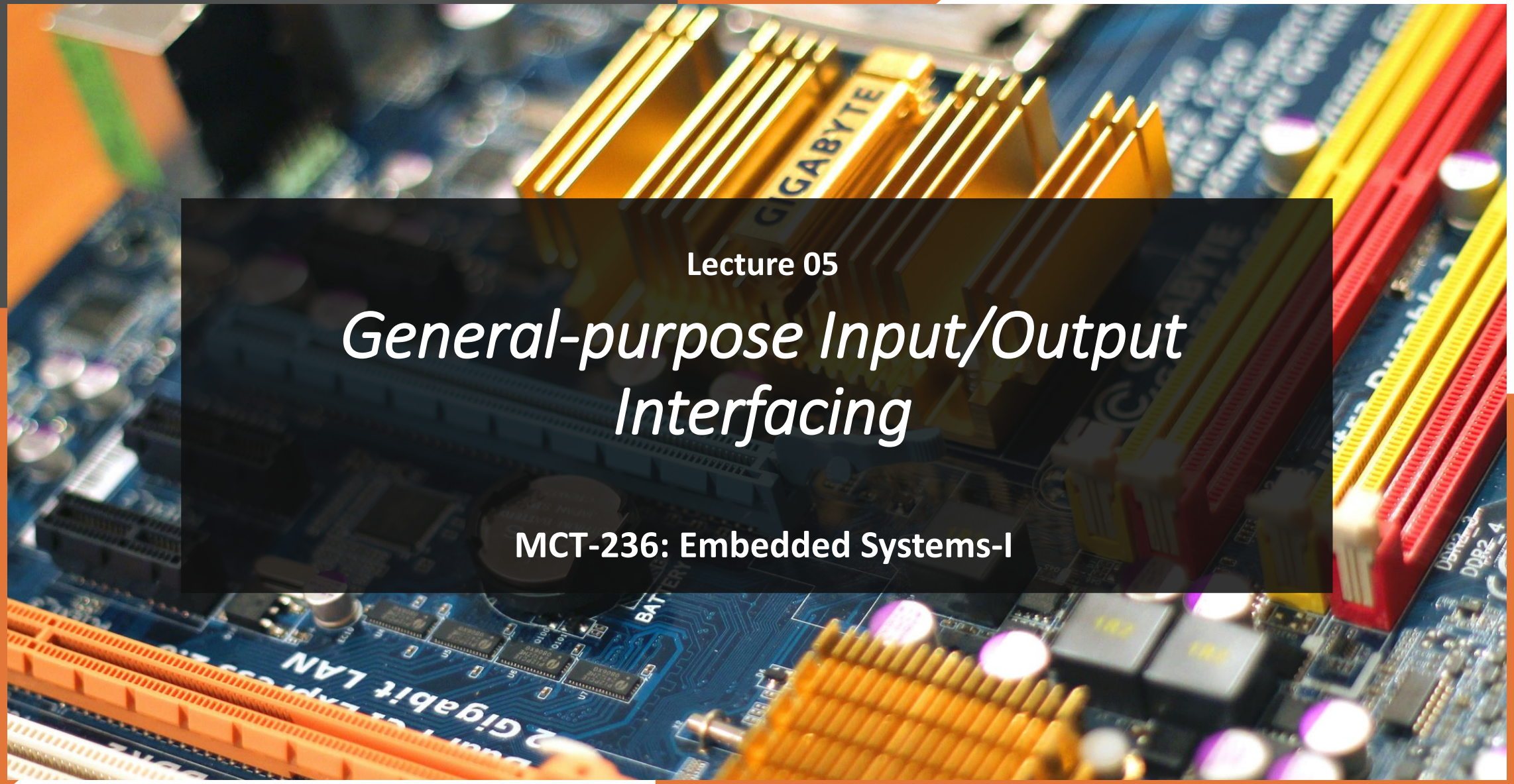
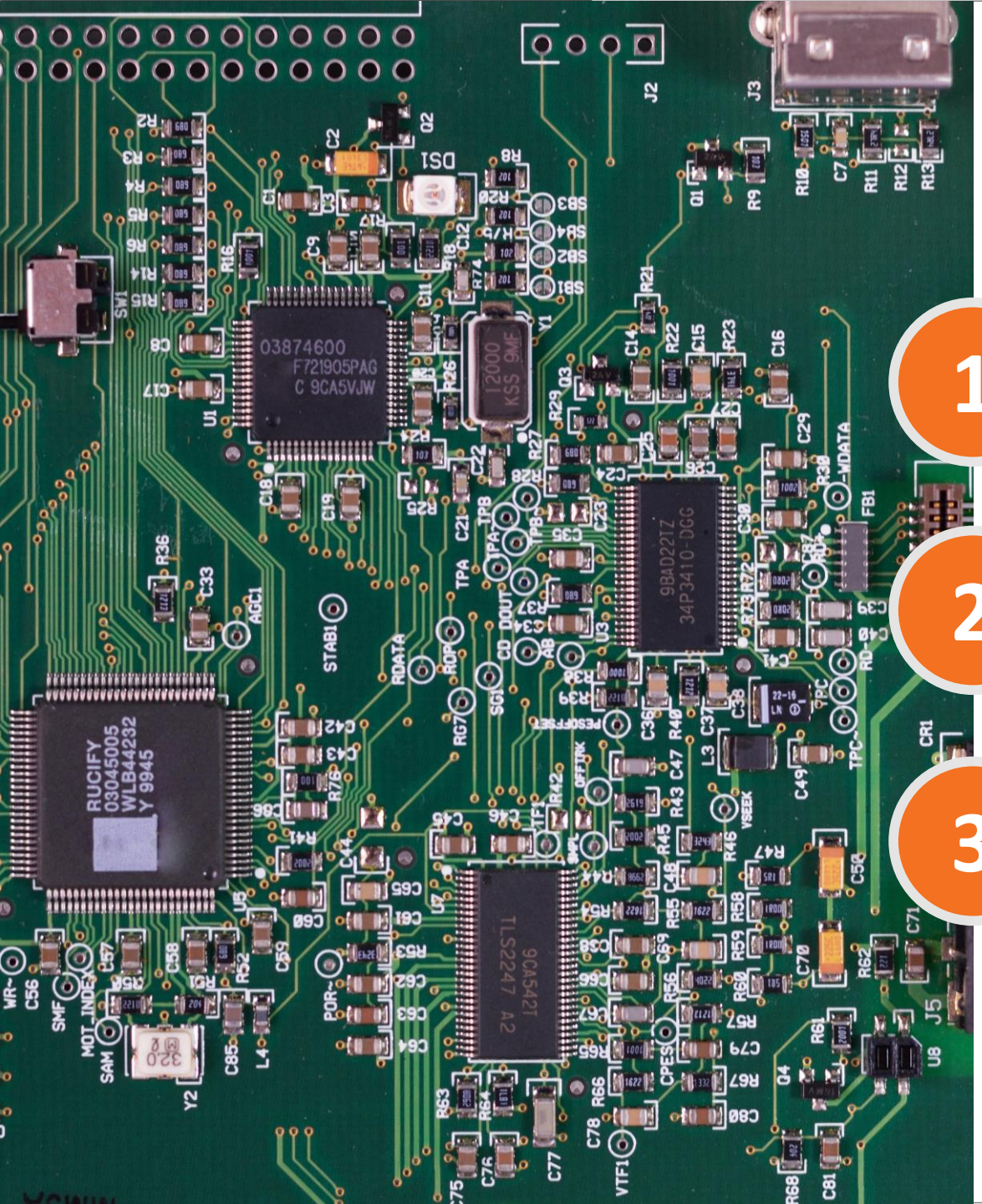


Lecture 05

General-purpose Input/Output Interfacing

MCT-236: Embedded Systems-I



1. BASICS OF GPIO INTERFACING
2. TM4C123 MICROCONTROLLER GPIOs
3. STEPS TO CONFIGURE MICROCONTROLLER PINS AS GPIO

BASICS OF GPIO INTERFACING

GPIO, Digital Output, Digital Input, and GPIO Features

GPIO

BASICS OF GPIO INTERFACING

- The microcontroller physical pins are highly flexible and can be configured either as an input or an output.
- Many of these input-output pins can also be configured for alternate functionalities
- A **general purpose input output** (GPIO) port is a physical pin in a microcontroller that can be configured by software to become either digital input or digital output.
- The collection of multiple digital GPIO pins is called a **parallel port**.

GPIO

BASICS OF GPIO INTERFACING

GPIO Pin as configured as Digital Output:

- allows to translate logical levels within the program to corresponding voltage levels on the associated pin by performing a write operation to the GPIO address
- the voltage levels on the pin configured as an output allow the microcontroller to exert an ON/OFF control to the device connected to that pin.

GPIO Pin as configured as Digital Input:

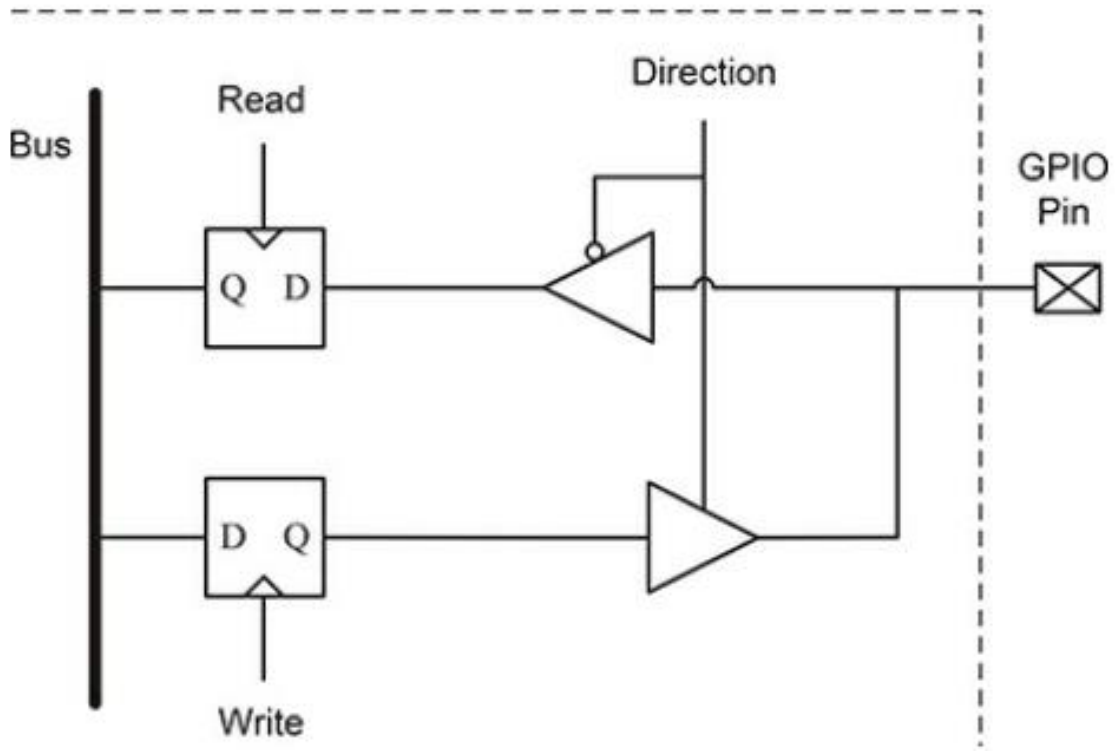
- the software can read external digital signals into the program to get the current state of the connected device.
- a read cycle access from the corresponding GPIO address returns the current state of the pin, which is configured as an input at that time.
- it is very important to ensure that the voltage levels of the external signal do not exceed the voltage tolerance levels of the GPIO pin.

GPIO Features

BASICS OF GPIO INTERFACING

GPIO Pin Direction Control:

- The purpose of this control is to configure a GPIO pin either input or output

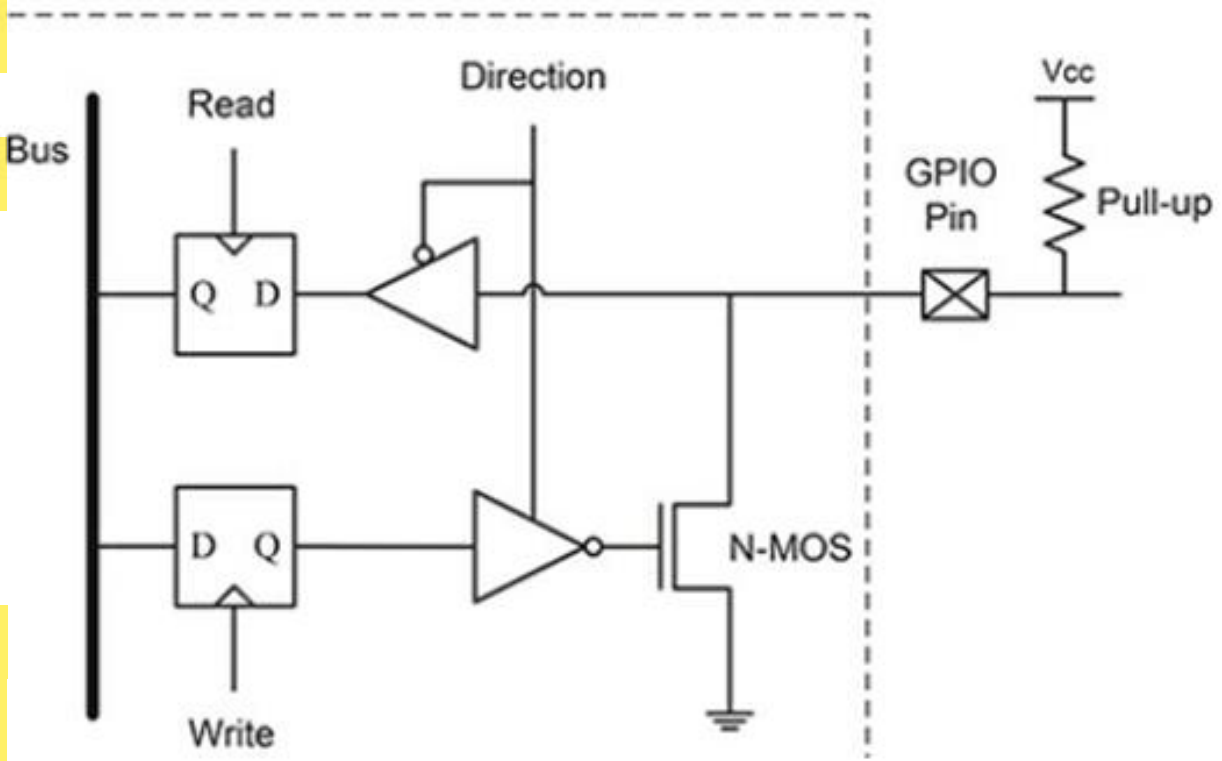


GPIO Features

BASICS OF GPIO INTERFACING

GPIO with Pull-up/Pull-down Resistors:

- Pull-up/pull-down resistor ensures that a configured GPIO pin is never in high-impedance state even if external interfaced device is disconnected
- Whenever external device becomes active, the voltage levels set by these resistors are overdriven by the corresponding logic level of the device output.
- Another possible use of pullup resistors is to interface a device with a microcontroller (or another device), when the operating supply voltages for them are different.



GPIO Features

BASICS OF GPIO INTERFACING

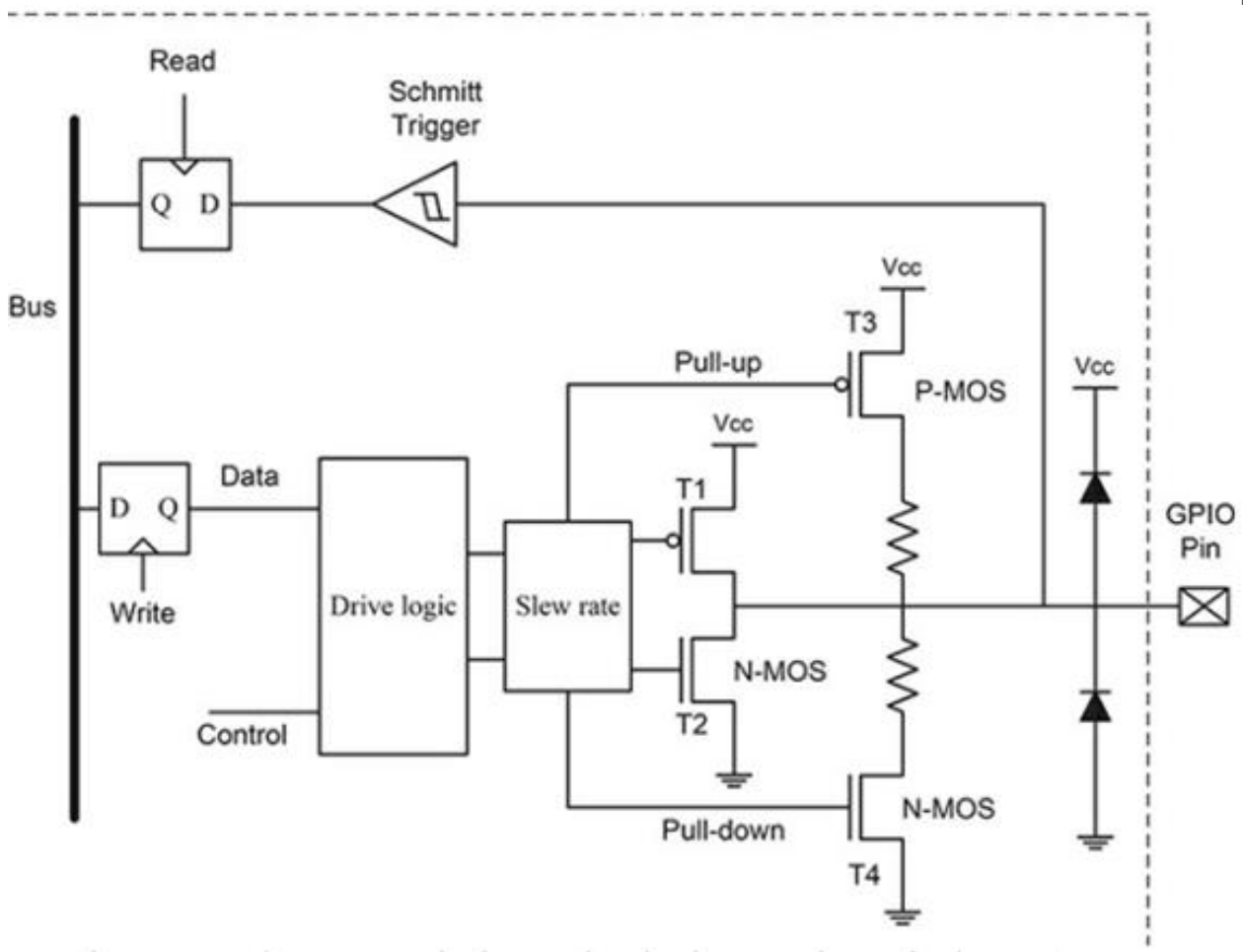
GPIO Current Sourcing and Sinking Capability:

- Each GPIO has limited current sourcing (in case of output) and sinking (in case of input) capability.
- Some of the complex GPIOs have multiple configurable levels of current sourcing capability.

GPIO as Input with Higher Voltage Tolerance:

- In some microcontrollers, GPIOs, when configured as inputs, can tolerate voltage levels higher than the operating supply voltage of the microcontroller.
- For instance, TM4C123 has GPIOs which can tolerate 5 V input signals despite its operating voltage of 3.3 V.

Some other important features are **slew rate control** for output configuration and **Schmitt triggered** type for inputs.



Multiplexing Functionalities on GPIO Pin

BASICS OF GPIO INTERFACING

- Most of the GPIO pins in a microcontroller can be configured for an alternate hardware function.
- It should be noted that only one of the alternate functions can be configured at a given time.
- The user program can switch among different alternate functionalities during execution

A GPIO pin may also be configured for any of the given alternate functionalities:

- Analog input
- UART/SPI/I2C/CAN/USB
- PWM
- Timer/CCP
- QEI
- others

TM4C123 MICROCONTROLLER GPIOs

GPIO Ports and their Bus Connectivity

Overview

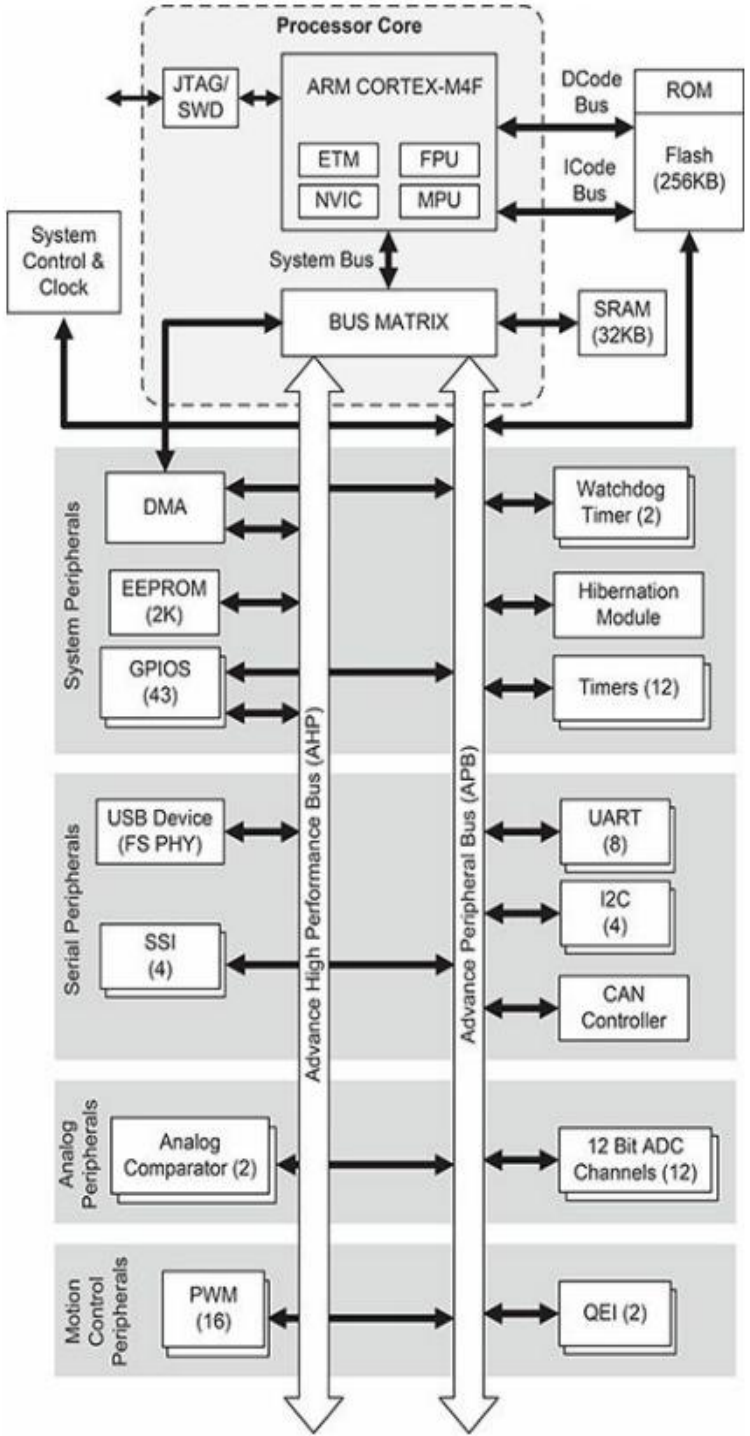
TM4C123 MICROCONTROLLER GPIOs

- Out of 64-pins of TM4C123GH6PM, 43 pins are GPIO
- Grouped in six labeled **PortA** to **PortF**
  - **PortA** to **PortD** are 8-pins ports, **PortE** is 6-pins, and **PortF** is 5-pins port
- Some of the port pins also have special peripheral functionalities multiplexed
- When configured as GPIO these port pins have the following capabilities
  - Internal weak pull-up or pull-down resistors
  - Each port pin can be configured as open drain
  - Slew rate control capability is provided for 8 mA output drive
  - Some of the port pins are capable of tolerating 5V when configured as inputs
  - Configurable current sourcing capability for levels of 2 mA, 4 mA, and 8 mA

Bus Connectivity

TM4C123 MICROCONTROLLER GPIOs

- ARM Cortex-M4 architecture uses memory-based peripherals
- Two on-chip busses that connect the processor core to the peripherals
  - **Advanced Peripheral Bus** (**APB**) is a low-speed legacy bus
  - **Advanced High-performance Bus** (**AHB**) is a high-speed bus
- GPIO module has connectivity available on both the buses

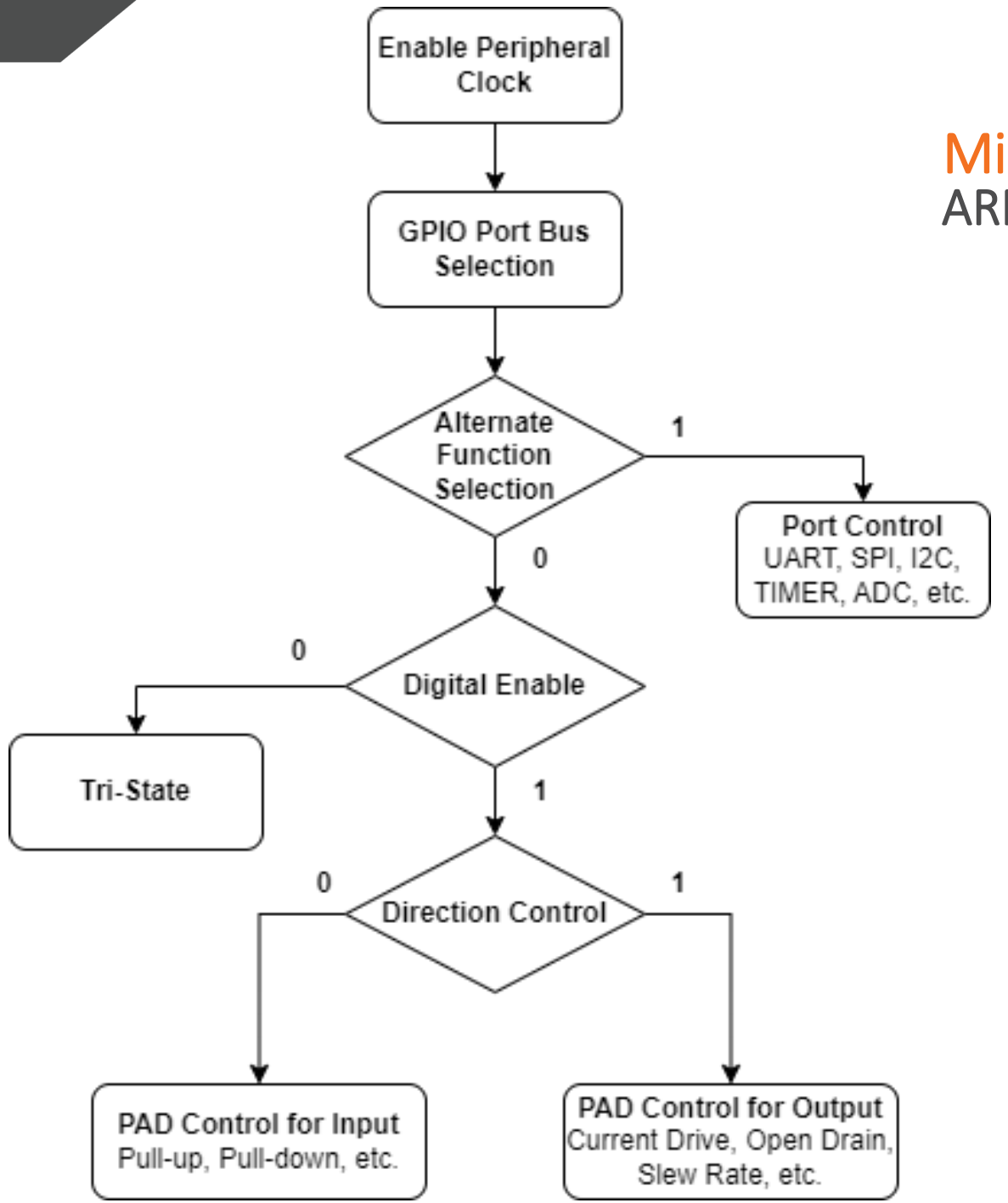


STEPS TO CONFIGURE MICROCONTROLLER PINS AS GPIO

Clock Configuration, GPIO Port Bus Selection, Mode Control, Pad Control, and Data Control Configuration

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller



Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Following are the basics steps required to configure microcontroller pin as GPIO:

- Step 1: Clock Configuration
- Step 2: GPIO Port Bus Selection
- Step 3: Mode Control Configuration
- Step 4: Pad Control Configuration
- Step 5: Data Control Configuration

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Step 1: Clock Enable

Step 2: GPIO Bus Selection

Step 3: Mode Control Config.

Step 4: Pad Control Config.

Step 5: Data Control Config.

- Enable clock for GPIO Port whether entire port or few pins are to be configured as GPIO
- **RCGC_GPIO_R** register, mapped to memory address 0x400FE608, is used enable clock for GPIO
- Bit 0 to 5 of **RCGC_GPIO_R** can be set to enable clock to **PortA** to **PortF**, respectively
  - e.g., 0010 0000 = 0x20 value written on **RCGC_GPIO_R** will enable clock to GPIO **PortF**
  - e.g., 0000 1001 = 0x09 value written on **RCGC_GPIO_R** will enable clock to GPIO **PortA** and **PortD**
- *After enabling the clock to a GPIO module, there must be a 3-clock cycle delay before accessing the GPIO registers*

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Step 1: Clock Enable

Step 2: GPIO Bus Selection

Step 3: Mode Control Config.

Step 4: Pad Control Config.

Step 5: Data Control Config.

- GPIO module or GPIO port registers can be accessed using two different buses. One of them is the legacy bus called Advanced Peripheral Bus (APB), while the other bus is the Advanced High-Performance Bus (AHB).
- All the registers associated with each port are accessible from both the buses, but they are mapped to different address spaces in the peripheral address space.
- For example, the registers associated with PortF on TM4C123 microcontroller are accessible from APB bus using the address space 0x40025000-0x40025FFF or they can be accessed from AHB bus using the address space 0x4005D000-0x4005DFFF.

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Step 1: Clock Enable

Step 2: GPIO Bus Selection

Step 3: Mode Control Config.

Step 4: Pad Control Config.

Step 5: Data Control Config.

- Bus configuration is selected (either **APB** or **AHB**) by selecting appropriate base address of the GPIO Port
  - e.g., 0x40025000 is the base address for **PortF** Registers APB bus

Base Address	Description
0x4000 4000	GPIO Port A Registers APB Bus
0x4000 5000	GPIO Port B Registers APB Bus
0x4000 6000	GPIO Port C Registers APB Bus
0x4000 7000	GPIO Port D Registers APB Bus
0x4002 4000	GPIO Port E Registers APB Bus
0x4002 5000	GPIO Port F Registers APB Bus
0x4005 8000	GPIO Port A Registers AHB Bus
0x4005 9000	GPIO Port B Registers AHB Bus
0x4005 A000	GPIO Port C Registers AHB Bus
0x4005 B000	GPIO Port D Registers AHB Bus
0x4005 C000	GPIO Port E Registers AHB Bus
0x4005 D000	GPIO Port F Registers AHB Bus

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Step 1: Clock Enable

Step 2: GPIO Bus Selection

Step 3: Mode Control Config.

Step 4: Pad Control Config.

Step 5: Data Control Config.

- GPIO Port pins can be configured for alternate functionalities e.g., PWM, ADC, SPI etc.
- When an alternate functionality is configured, port pins cannot be used as GPIO
- GPIO Alternate Function Select Register (**GPIO_AFSEL_R**, Offset value **0x420**) is used for configuring alternate functionality
- This register must be cleared to use GPIO functionality
- e.g., to disable alternate functionality on all the pins of GPIO **PortA** (base address: 0x40004000), we will write 0x00 on register 0x40004420 (= 0x40004000 + 0x420)

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Step 1: Clock Enable

Step 2: GPIO Bus Selection

Step 3: Mode Control Config.

Step 4: Pad Control Config.

Step 5: Data Control Config.

- To configure a specific alternate functionality, the GPIO Port Control (**GPIO_PCTL_R**) register is used, which selects one of several available peripheral functions multiplexed for the specific microcontroller pin.
- It is also possible to use some of the pins for analog input functionality. When a port pin is to be used as an analog to digital converter (ADC) input, the appropriate bit in the GPIO Analog Mode Select (**GPIO_AMSEL_R**) register must be set to disable the analog isolation circuit.

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Step 1: Clock Enable

Step 2: GPIO Bus Selection

Step 3: Mode Control Config.

Step 4: Pad Control Config.

Step 5: Data Control Config.

- Multiple Pad Control registers are required to configure
  - Digital Enable: **GPIO_DEN_R** (Offset Value: **0x51C**)
  - Pull Up Configuration: **GPIO_PU_R** (Offset Value: **0x510**)
  - Pull Down Configuration: **GPIO_PD_R** (Offset Value: **0x514**)
  - Slew Rate Configuration: **GPIO_SL_R** (Offset Value: **0x518**)
- Each GPIO Pin can be individually configured using these Pad Control registers
- To use a pin as GPIO, the corresponding bit in **GPIO_DEN_R** must be set
- e.g. if we want to enable pin 3 and 4 of GPIO **PortF** (base address: 0x40025000), we will write 0x18 (0001 1000) on register **GPIO_DEN_R** 0x4002551C (= 0x40025000 + 0x51C)

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Step 1: Clock Enable

Step 2: GPIO Bus Selection

Step 3: Mode Control Config.

Step 4: Pad Control Config.

Step 5: Data Control Config.

PIN DIRECTION CONTROL

- Each individual port pin can be configured as an Input or Output using register **GPIO_DIR_R** (Offset Value: **0x400**)
- Clearing a register bit configures corresponding port pin as an Input and setting a register bit configures corresponding port pin as an Output
- e.g. if we want to configure GPIO **PortF** (base address: 0x40025000) pin 3 as output and 4 as input, we will write 0x08 (0000 1000) on register **GPIO_DIR_R** 0x40025400 (= 0x40025000 + 0x400)

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Step 1: Clock Enable

Step 2: GPIO Bus Selection

Step 3: Mode Control Config.

Step 4: Pad Control Config.

Step 5: Data Control Config.

GPIO MASKING

- **GPIO_DATA_R** (Offset Value: **0x000-0x3FC**) is used to read/write data from/to GPIO pins
- When configured as input, the value of GPIO pin is captured and stored in corresponding bit of **GPIO_DATA_R** register
- When configured as output, the corresponding **GPIO_DATA_R** register bit value is drive the GPIO port pin
- Some port pins can be configured as input while others as output and **GPIO_DATA_R** can be used to read (input data) and write (output data) operations

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Step 1: Clock Enable

Step 2: GPIO Bus Selection

Step 3: Mode Control Config.

Step 4: Pad Control Config.

Step 5: Data Control Config.

GPIO MASKING

- In a single instruction cycle, read or write operation for individual (or multiple) GPIO port pins can be performed without affecting others
- Bit 2 to 9 can be used to construct the mask and corresponding 12-bit value is used as Offset value
- e.g. if we want to access only pin 1 of GPIO **PortF** (base address: **0x40025000**) then 0x008 (0000 0000 1000) will be the Offset value and we will perform operation on **GPIO_DATA_R** located at 0x40025000 (= 0x40025008 + 0x008)

Steps to Configure Microcontroller Pins as GPIO

ARM Cortex-M4 Microcontroller

Step 1: Clock Enable

Step 2: GPIO Bus Selection

Step 3: Mode Control Config.

Step 4: Pad Control Config.

Step 5: Data Control Config.

Offset Address	Offset Address Bits	Accessible Bits
0x000	0000 0000 0000	No bit accessible
0x004	0000 0000 0100	Bit 0 is accessible
0x008	0000 0000 1000	Bit 1 is accessible
0x00C	0000 0000 1100	Bit 0 and 1 are accessible
0x010	0000 0001 0000	Bit 2 is accessible
0x01C	0000 0001 1100	Bit 0, 1, and 2 are accessible
0x020	0000 0010 0000	Bit 3 is accessible
0x030	0000 0011 0000	Bit 2 and 3 are accessible
0x038	0000 0011 1000	Bit 1, 2, and 3 are accessible
…	…	…
0x3F8	0011 1111 1000	Bit 1 to 7 are accessible
0x3FC	0011 1111 1100	Bit 0 to 7 are accessible

THANK YOU

Any Questions???