GPIO : GENERAL PURPOSE INPUT OUTPUT

MUHAMMAD ELZEINY

FEATURES

- Up to 43 GPIOs
- Flexible pin muxing allows use as GPIO or one of several peripheral function
- 5V tolerant in input configuration.
- Programmable control for GPIO External interrupts.
- Bit masking in both read and write operations through address lines
- Pins configured as digital inputs are Schmitt-triggered
- Programmable control for GPIO pad
    - Internal pull up, pull down resistor and Open drain enables
    - 2-mA, 4-mA, and 8-mA pad drive current
    - Slew rate control for 8-mA pad drive

SIGNAL DESCRIPTION

- GPIO signals have alternate hardwarefunctions.
- The table (10-2)lists the GPIO pins and their analog and digital alternate functions.
- All GPIO signals are 5-V tolerant when configured as inputs except for PD4, PD5, PB0 and PB1, which are limited to 3.6 V.

Functional description

1. Data Control

- Data Direction Operation
- Data Register Operation
  - address bus as a mask method

2. Interrupt Control

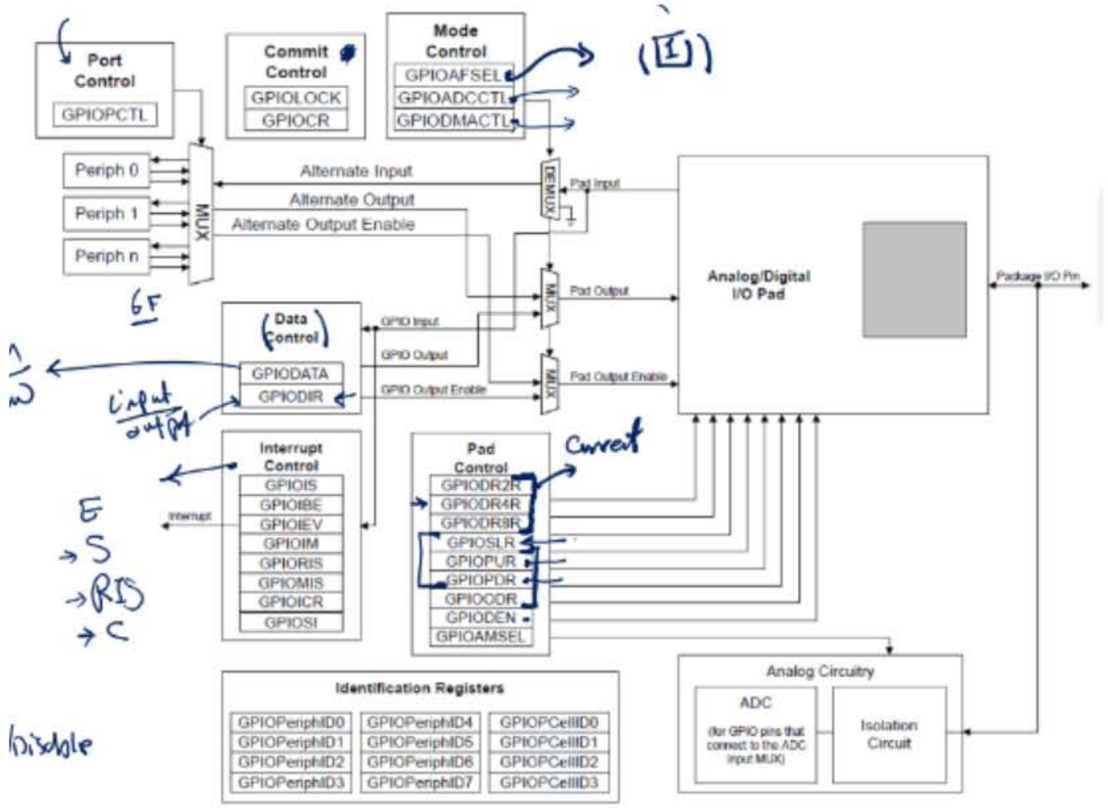
3. Mode Control

4. Commit Control

The GPIO commit control registers provide a layer of protection against accidental programming of critical hardware peripherals.

5. Pad Control

These registers control drive strength, open -drain configuration, pull -up and pull -down resistors, slew rate control and digital input enable for each GPIO.



#TASK : PORT DRIVER IMPLEMENTATION

API – Types

- Port_PinType
- Port_PinDirectionType
- Port_PinModeType
- Port_PinInternalAttachType
- Port_PinOutputCurrentType
- Port_ConfigType

API - Functions

- void Port_Init (const Port_ConfigType* ConfigPtr)

Configuration

- PortPinMode
- PortPinLevelValue
- PortPinDirection
- PortPinInternalAttach
- PortPinOutputCurrent

#TASK : DIO DRIVER IMPLEMENTATION

API - Types

- Dio_ChannelType
- Dio_PortType
- Dio_LevelType
- Dio_PortLevelType

API – Functions

- Dio_LevelType Dio_ReadChannel (Dio_ChannelType ChannelId)
- void Dio_WriteChannel (Dio_ChannelType ChannelId, Dio_LevelType Level)
- Dio_PortLevelType Dio_ReadPort (Dio_PortType PortId)
- void Dio_WritePort (Dio_PortType PortId, Dio_PortLevelType Level)
- Dio_LevelType Dio_FlipChannel (Dio_ChannelType ChannelId)

• Configurations

- There is no required configurations

API : FUNCTIONS

- void Gpt_Init (const Gpt_ConfigType* ConfigPtr)
  - The function Gpt_Init shall initialize the hardware timer module according to a configuration set referenced by ConfigPtr
  - The function Gpt_Init shall disable all interrupt notifications, controlled by the GPT driver
  - The function Gpt_Init shall set the operation mode of the GPT driver to "normal mode".
  - The function Gpt_Init shall start all enabled GPT Predef Timers at value “0”.
- void Gpt_DisableNotification (Gpt_ChannelType Channel)
- void Gpt_EnableNotification (Gpt_ChannelType Channel)

API : FUNCTIONS

- Gpt_ValueType Gpt_GetTimeElapsed (Gpt_ChannelType Channel)
  - Synchronous, Reentrant
  - Consider the special cases stated in the table.
- Std_ReturnType Gpt_GetPredefTimerValue (Gpt_PredefTimerType PredefTimer, uint32*TimeValuePtr)
  - Synchronous, Reentrant
  - Shall return the current value of the GPT Predef Timer passed by PredefTimer.
- Gpt_ValueType Gpt_GetTimeRemaining (Gpt_ChannelType Channel)
  - Returns the time remaining until the target time is reached.
  - Synchronous, Reentrant

API : FUNCTIONS

- void Gpt_StartTimer (Gpt_ChannelType Channel, Gpt_ValueType Value)
  - Synchronous, Reentrant(but not for the same timer channel)
  - shall start the selected timer channel with a defined target time.
- void Gpt_StopTimer (Gpt_ChannelType Channel)
  - Synchronous, Reentrant(but not for the same timer channel)
- void Gpt_Notification_<channel> (void)
  - The callback notifications Gpt_Notification_<channel> shall be configurable as pointers to user defined functions within the configuration structure

POINTS TO BE DEFINES (HW SPECIFIC)

- How to Set timer Mode (one-shot \ Continuous) **GPTMTnMR**
- Mandatory Init Hw parametes (Timer Resolution : **GPTMCFG)**
- How to Start\Stop Counting **GPTMCTL**
- How to Enable\Disable Interrupt **(GPTMIMR)**
- How to get elapsed\remaining time value (Ticks) **(GPTMTAV)**
- How to Set TickFreq