GPIO 接口函数

*1. u8 FGpioPs_init(FGpioPs_T *dev, u32 addr)*

描述	* This function initial GPIO structure
参数	* @param dev is gpio device handle. * @param instance is a pointer to GpioX instance.
返回值	* @return 0 if successful, otherwise 1.

*2. u8 FGpioPs_writeData(FGpioPs_T *dev, u32 data)*

描述	* This function write data to GPIO port
参数	* @param dev is gpio device handle. * @param data is the data of data write to GPIO port
返回值	* @return 0 if successful, otherwise 1.

*3. u32 FGpioPs_readData(FGpioPs_T *dev)*

描述	* This function read data from GPIO port
参数	* @param dev is gpio device handle.
返回值	* @return return gpio prot data.

*4. u32 FGpioPs_readData(FGpioPs_T *dev)*

描述	* This function returns the data register value.
参数	* @param dev is gpio device handle.
返回值	* @return return gpio prot data.

*5. u32 FGpioPs_getExtPort(FGpioPs_T *dev)*

描述	* This function returns the value of external port data register.
参数	* @param dev is gpio device handle.
返回值	* @return data of extern PORT data

*6. u8 FGpioPs_writeBit(FGpioPs_T *dev, enum FGpioPs_state value, u32 bits)*

描述	* This function sets the specified bits of a port data register to the * value specified. Multiple bits can be specified in the function arguments, * using the bitwise 'OR' operator. The allowable values that a port bit can * be set to are 0 and 1
参数	* @param dev is gpio device handle. * @param value is value to set(Gpio_low/Gpio_high). * @param bits -- port bits to change.
返回值	* @return 0 if successful, otherwise 1.

*7. enum FGpioPs_state FGpioPs_getBit(FGpioPs_T *dev, enum FGpioPs_bit bit)*

描述	* This function returns the value of a port data register bit. Only * one bit may be specified per invocation of this function.
参数	* @param dev is gpio device handle. * @param bit -- bit to return the value of
返回值	* Gpio_high -- if the specified port bit is set (0x1) * Gpio_low -- if the specified port bit is clear (0x0)

*8. u8 FGpioPs_setDirection(FGpioPs_T *dev, u32 data)*

描述	* This function set GpioX direction
参数	* @param dev is gpio device handle. * bitX = 0 -- input; bitX = 1 -- output.
返回值	* @return 0 if successful, otherwise 1.

*9. u32 FGpioPs_getDirection(FGpioPs_T *dev)*

描述	* This function get GpioX direction
参数	* @param dev is gpio device handle. * bitX = 0 -- input; bitX = 1 -- output.
返回值	* @return 0 if successful, otherwise 1.

*10. u32 FGpioPs_setBitDirection(FGpioPs_T *dev, u32 bits, enum FGpioPs_direction direction)*

描述	* This function sets the specified bits of a port data direction * register to the value specified. Multiple bits of a single port can * be specified in the function arguments, using the bitwise 'OR' * operator. The allowable values that a port data direction bit can * be set to are Gpio_input and Gpio_output.
参数	* @param dev is GpioX device handle. * @param bits -- port bits to change. * @param direction is value to set(Gpio_input/Gpio_output).
返回值	* @return 0 if successful, otherwise 1.

*11. enum FGpioPs_direction FGpioPs_getBitDirection(FGpioPs_T *dev, enum FGpioPs_bit bit)*

描述	* This function returns the value of a port data direction bit. Only * one bit may be specified per invocation of this function.
参数	* @param dev is GpioX device handle. * @param bit -- port bit to check
返回值	* @return * Gpio_input -- if the port bit is an input * Gpio_output -- if the port bit is an output * Gpio_no_direction -- if the specified port bit is not available

*12. u32 FGpioPs_enableIrq(FGpioPs_T *dev, u32 interrupts)*

描述	* This function enables interrupts for the specified bit(s) of port * Multiple bits can be specified in the function arguments, using the * bitwise 'OR' operator.
参数	* @param dev is gpio device handle. * @param interrupts -- interrupt bit(s) to enable
返回值	* @return 0 if successful, otherwise 1.

*13. u32 FGpioPs_disableIrq(FGpioPs_T *dev, u32 interrupts)*

描述	* This function disables interrupts for the specified bit(s) of port * . Multiple bits can be specified in the function arguments, using * the bitwise 'OR' operator.
参数	* @param dev is gpio device handle. * @param interrupts -- interrupt bit(s) to enable

返回值	* @return 0 if successful, otherwise 1.

*14. u8 FGpioPs_maskIrq(FGpioPs_T *dev, u32 interrupts)*

描述	* This function masks interrupts for the specified bit(s) of port. * Multiple bits can be specified in the function arguments, using the * bitwise 'OR' operator.
参数	* @param dev is gpio device handle. * @param interrupts -- interrupt bit(s) to mask
返回值	* @return 0 if successful, otherwise 1.

*15. u8 FGpioPs_unmaskIrq(FGpioPs_T *dev, u32 interrupts)*

描述	* This function unmasks interrupts for the specified bit(s) of port. * Multiple bits can be specified in the function arguments, using the * bitwise 'OR' operator.
参数	* @param dev is gpio device handle. * @param interrupts -- interrupt bit(s) to unmask
返回值	* @return 0 if successful, otherwise 1.

*16. BOOL FGpioPs_isIrqMasked(FGpioPs_T *dev, enum FGpioPs_bit interrupt)*

描述	* This function returns whether interrupts are masked for a * particular bit of port or not. Only one bit may be specified per * invocation of this function.
参数	* @param dev is gpio device handle. * @param interrupt is bit of port A to check

返回值	* TRUE -- if interrupts are masked for the specified bit * FALSE -- if interrupts are not masked for the specified bit

*17. u32 FGpioPs_getIrqMask(FGpioPs_T *dev)*

描述	* This function returns the value of the interrupt mask register.
参数	* @param dev is gpio device handle.
返回值	* The interrupt mask register value.

*18. u32 FGpioPs_setIrqType(FGpioPs_T *dev, enum FGpioPs_irq_type type,*

u32 interrupts)

描述	* This function sets the interrupt type for the specified bit(s) of * port. There are two types of interrupts available, * level-sensitive and edge-sensitive. The polarity of these interrupt * types is set using dw_gpio_setIrqPolarity(). Multiple bits can be * specified in the function arguments, using the bitwise 'OR' * operator.
参数	* @param dev is gpio device handle. * @param type -- interrupt type to set * @param interrupts -- interrupt bit(s) to set
返回值	* @return 0 if successful, otherwise 1.

*19. enum FGpioPs_irq_type dw_gpio_getIrqType(FGpioPs_T *dev, enum FGpioPs_bit interrupt)*

描述	* This function returns the interrupt type for a specified but of * port . Only one bit may be specified per invocation of this * function.
参数	* @param dev is gpio device handle. * @param interrupt -- bit of port to check
返回值	* Gpio_level_sensitive -- if the interrupt bit is level-sensitive * Gpio_edge_sensitive -- if the interrupt bit is edge-sensitive * Gpio_no_type -- if interrupts are not supported or the * specified bit is not available

*20. u32 FGpioPs_setIrqPolarity(FGpioPs_T *dev, enum FGpioPs_irq_polarity polarity, u32 interrupts)*

描述	* This function sets the interrupt polarity for the specified bit(s) * of port. The polarity can be either active-low or active-high. * For edge-sensitive interrupts, active-low corresponds to a * falling-edge interrupt while active-high corresponds to a * rising-edge interrupt. The interrupt type is set using * FGpioPs_setIrqType(). Multiple bits can be specified in the * function arguments, using the bitwise 'OR' operator.
参数	* @param dev is gpio device handle. * @param polarity -- interrupt polarity to set * @param interrupts -- interrupt bit(s) to set

返回值	* @return 0 if successful, otherwise 1.

*21. enum FGpioPs_irq_polarity FGpioPs_getIrqPolarity(FGpioPs_T *dev,*

enum FGpioPs_bit interrupt)

描述	* This function returns the interrupt polarity of a specified bit of * . Only one bit may be specified per invocation of this function.
参数	* @param dev is gpio device handle. * @param polarity -- interrupt polarity to set * @param interrupts -- bit of port to check
返回值	* Gpio_active_low -- if interrupt bit is active-low/falling-edge * Gpio_active_high -- if interrupt bit is active-high/rising-edge * Gpio_no_polarity -- if interrupts are not supported

*22. u32 FGpioPs_getActiveIrq(FGpioPs_T *dev)*

描述	* This function returns the value of the interrupt status register.
参数	* @param dev is gpio device handle.
返回值	* @return The interrupt status register value.

*23. u32 FGpioPs_clearIrq(FGpioPs_T *dev, u32 interrupts)*

描述	* This function is used to clear edge-sensitive interrupts of port. * Multiple bits can be specified in the function arguments, using the * bitwise 'OR' operator.
参数	* @param dev is gpio device handle. * @param interrupts is interrupt bit(s) to clear

返回值	* @return 0 if successful, otherwise 1.

*24. u32 FGpioPs_enableDebounce(FGpioPs_T *dev, u32 interrupts)*

描述	* This function enables debounce logic for the specified bit(s) of * port. When enabled, a signal must be valid for two periods of the * external debounce clock before it is internally processed. * Multiple bits can be specified in the function arguments, * using the bitwise 'OR' operator.
参数	* @param dev is gpio device handle. * @param interrupts -- interrupt bit(s) to set
返回值	* @return 0 if successful, otherwise 1.

*25. u32 FGpioPs_disableDebounce(FGpioPs_T *dev, u32 interrupts)*

描述	* This function disables debounce logic for the specified bit(s) of * port A. Multiple bits can be specified in the function arguments, * using the bitwise 'OR' operator.
参数	* @param dev is gpio device handle. * @param interrupts -- interrupt bit(s) to set
返回值	* @return 0 if successful, otherwise 1.

*26. BOOL FGpioPs_isDebounceEnabled(FGpioPs_T *dev, enum FGpioPs_bit interrupt)*

描述	* This function returns whether debounce is enabled for the specified * it or port or not. Only one bit may be specified per invocation * of this function.

参数	* @param dev is gpio device handle. * @param interrupt -- bit of port to check
返回值	* @return * TRUE -- if debounce is enabled for interrupt bit * FALSE -- if debounce is disabled for interrupt bit

*27. u32 FGpioPs_enableSync(FGpioPs_T *dev)*

描述	* This function enables interrupt synchronization. When enabled, all * level-sensitive interrupts are synchronized to pclk.
参数	* @param dev is gpio device handle.
返回值	* @return 0 if successful, otherwise 1.

*28. u32 FGpioPs_disableSync(FGpioPs_T *dev)*

描述	* This function disables synchronization for level-sensitive * interrupts.
参数	* @param dev is gpio device handle.
返回值	* @return 0 if successful, otherwise 1.

*29. u32 FGpioPs_isSynced(FGpioPs_T *dev)*

描述	* This function returns whether synchronization is enabled for * level-sensitive interrupts or not.
参数	* @param dev is gpio device handle.
返回值	* @return * TRUE -- if interrupt synchronization is enabled

	* FALSE -- if interrupt synchronization is not enabled

*30. u8 FGpioPs_enableIrqBothEdge(FGpioPs_T *dev, u32 interrupts)*

描述	* This function enable both edge interrupt, when this function is called * , gpio level and polarity is no sense.
参数	* @param dev is gpio device handle. * @param interrupts -- interrupt bit(s) to set
返回值	* @return 0 if successful, otherwise 1.

*31. u8 FGpioPs_disableIrqBothEdge(FGpioPs_T *dev, u32 interrupts)*

描述	* This function disable both edge interrupt, when this function is called * , interrupt occurs depending on gpio level and polarity .
参数	* @param dev is gpio device handle. * @param interrupts -- interrupt bit(s) to set
返回值	* @return 0 if successful, otherwise 1.

*32. u32 FGpioPs_getBothEdgeInt(FGpioPs_T *dev)*

描述	* This function returns value of both edge interrupt reg.
参数	* @param dev is gpio device handle.
返回值	* @return value of both edge interrupt reg.

*33. u32 FGpioPs_getVerIdCode(FGpioPs_T *dev)*

描述	* This function get gpio version ID code.
参数	* @param dev is gpio device handle.

返回值	* @return version ID code.

*34. u32 FGpioPs_getIdCode(FGpioPs_T *dev)*

描述	* This function get gpio ID code.
参数	* @param dev is gpio device handle.
返回值	* @return ID code.