Freescale Semiconductor, Inc.
Application Note

Document Number: AN5078
Rev. 0, 02/2015

Influence of Pin Setting on System Function and Performance

by: Jiri Kotzian

Contents

1 Introduction

The purpose of this user’s guide is to provide details about setting functional and electrical parameters of the MCU and MPU pins. The pin configuration is complex to grasp at first. The description of each feature should help to set each pin correctly for any application.

2 Pins and pads

A pin represents a physical input or output carrying an electrical signal. Every input or output signal goes through a physical pin from or into a component. The term “pin” is used especially in schematics. Contact pads are designated surface areas of a printed circuit board or die of an integrated circuit. A pin of a component is soldered to a pad. An example is shown in Figure 1.

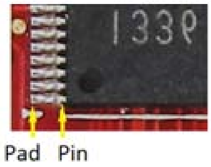


Figure 1. Pin and pad

The term “pin” should be used for components and the term “pad” should be used for PCBs. Since pins become pads on the BGA components, both terms are mixed. Moreover, on the BGA packages, the term “ball” is used. The term “ball” indicates a ball of soldering material attached to every pad during the manufacture process, usually from the bottom side of a component – see Figure 2.

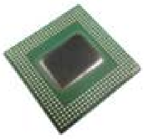


Figure 2. Bottom side of the BGA package

Every pin can serve as a signal, clock, power supply, or power reference. The physical pin name depends on the device package used.

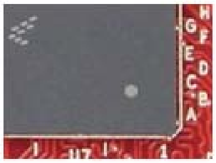


Figure 3. Top side of the BGA package with pin name notes

- In the 364 BGA package, the pins are named as a combination of a letter and a number, for example, A1-Y20 (ball). The first pad is A1 – see Figure 3.
- In the 176 LQFP package, the pins are numbered as 1 – 176 (pin). The first pin is 1.

Logical / canonical pin name is a unique alphanumeric identifier. The name depends on the meaning of the signal, for example:

- PTB0, which means port B pin 0
- PTA0 – PTA31, which means port A, pins 0 to 31
- PTF0 – PTF15, which means port F, pins from 0 to 15
- USB0_VBUS_DETECT, which means USB module 0, signal VBUS_DETECT

One pin usually has more than one function, called alternate functions. Each pin can have many alternate logical pin names. In the schematic of a symbol we can usually see a list of those features separated by the " / " symbol.

The net name is used in schematics to interconnect pins of different components. The net name is usually the same as or similar to the logical pin name. A wire in a schematic bear a "net name" assigned to it, for example:

- PTB0
- PTB10
- RESET_B

2.1 Example

Table 1. Assign pad, pin and alternate function in a datasheet

364 MAP BGA	176 LQFP	Pin name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
T6	49	PTB0	–	PTB0	FTM0_CH0	ADC0_SE2	TRACE CTL	LCD34	SAI2_RX_BCLK	VIU_DATA18	QSPI1_A_CS0	–

For example, the BGA package with 364 pads and the component datasheet information in Table 1. The physical pad T6 in the BGA package has the logical pad name PTB0.

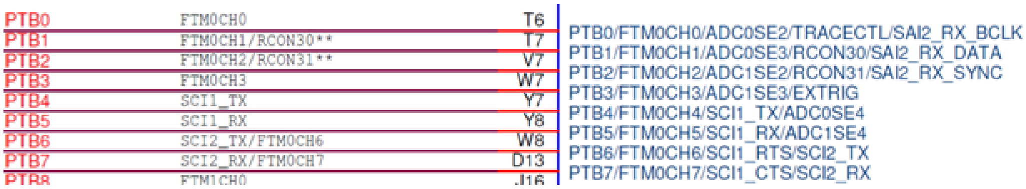

PTB0
FTM0CH0
T6
PTB1
FTM0CH1/RCON30**
T7
PTB2
FTM0CH2/RCON31**
V7
PTB3
FTM0CH3
W7
PTB4
SCI1_TX
Y7
PTB5
SCI1_RX
Y8
PTB6
SCI2_TX/FTM0CH6
W8
PTB7
SCI2_RX/FTM0CH7
D13
PTB8
FTM1CH0
J16
PTB0/FTM0CH0/ADC0SE2/TRACECTL/SAI2_RX_BCLK
PTB1/FTM0CH1/ADC0SE3/RCON30/SAI2_RX_DATA
PTB2/FTM0CH2/ADC1SE2/RCON31/SAI2_RX_SYNC
PTB3/FTM0CH3/ADC1SE3/EXTRIG
PTB4/FTM0CH4/SCI1_TX/ADC0SE4
PTB5/FTM0CH5/SCI1_RX/ADC1SE4
PTB6/FTM0CH6/SCI1_RTS/SCI2_TX
PTB7/FTM0CH7/SCI1_CTS/SCI2_RX

Figure 4. Assign pad, pin and alternate function in a schematic

The schematic snip in Figure 4 shows a connection of an MCU on the right side with wire nets on the left side. For example, the physical pad T6 in the BGA package has logical pad names (logical pad name list in blue) PTB0 / FTM0CH0 / ADC0SE2 / TRACECTL / SAI2_RX_BCLK and can serve as GPIO PTB0 (ALT0), FlexTimer channel 0 FTM0_CH0 (ALT1) or ADC input ADC0_SE2 (ALT2) and so on. The pad is connected to the net with net name PTB0 (in red).

NOTE

The FTM0CH0 (in black) indicates which alternate function of the pad is used in the application.

3 Pin features

Each pin of an MCU can be connected to one of the MCU's integrated peripherals, but it also has dedicated electrical features and characteristics.

3.1 Alternate functions

Each pin of an MCU can be used for different purposes. For example, the same pin can serve as a general purpose input / output (GPIO) or as a display controller output. The function selected depends on the pin setting in the appropriate multiplexer.

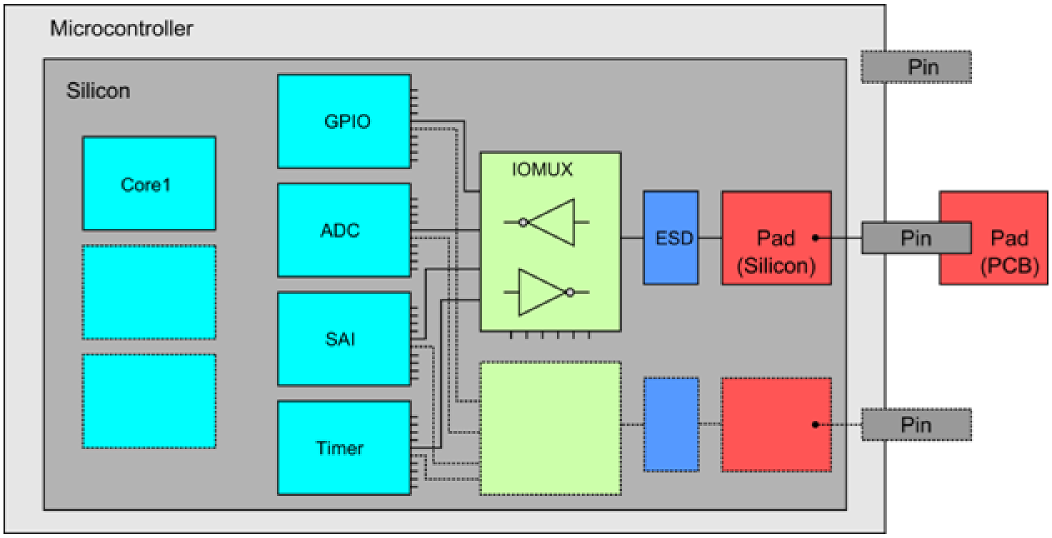


Figure 5. Alternate functions of a pin

A simplified block diagram is shown in Figure 5. One pin can work as GPIO, ADC input, SAI, or timer input / output.

3.2 Pin setting

While on small MCUs there is usually one register to set the whole port, each pin has dedicated register(s) on advanced MCUs and MPUs. Therefore, an MCU with six 32-bit ports may include 192 registers to set all pins.

Table 2. Pin setting registers

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value
4004_8058	Software MUX Pad Control Register 22 (IOMUXC_PTB0)	32	R / W	0000_0060h
4004_805C	Software MUX Pad Control Register 23 (IOMUXC_PTB1)	32	R / W	0030_0060h
4004_8060	Software MUX Pad Control Register 24 (IOMUXC_PTB2)	32	R / W	0030_0060h
4004_8064	Software MUX Pad Control Register 25 (IOMUXC_PTB3)	32	R / W	0000_0060h

Table 2. Pin setting registers (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value
4004_8068	Software MUX Pad Control Register 26 (IOMUXC_PTB4)	32	R / W	0000_0060h
4004_806C	Software MUX Pad Control Register 27 (IOMUXC_PTB5)	32	R / W	0000_0060h
4004_8070	Software MUX Pad Control Register 28 (IOMUXC_PTB6)	32	R / W	0000_0060h
4004_8074	Software MUX Pad Control Register 29 (IOMUXC_PTB7)	32	R / W	0000_0060h

All those registers form a separate peripheral with different names depending on a specific component; for example, named as IOMUX controller (IOMUXC), used by Vybrid and i.MX families – see Table 2.

3.3 Internal structure of IOMUX

The IOMUX allows an MCU to share one pad for several functional blocks. This sharing is done by multiplexing pad's input and output signals. Every module requires a specific pad setting, and for each pad there are up to eight muxing options (called ALT modes as seen in Section 3.1, "Alternate functions"). From a pin direction point of view, the IOMUX consists of three components:

- Input (set by IBE and HYS in IOMUXC)
- Output (set by OBE, ODE, DSE, SRE, and SPEED in IOMUXC)
- Pull-up / pull-down / keeper (set by PKE, PUE and PES in IOMUXC)

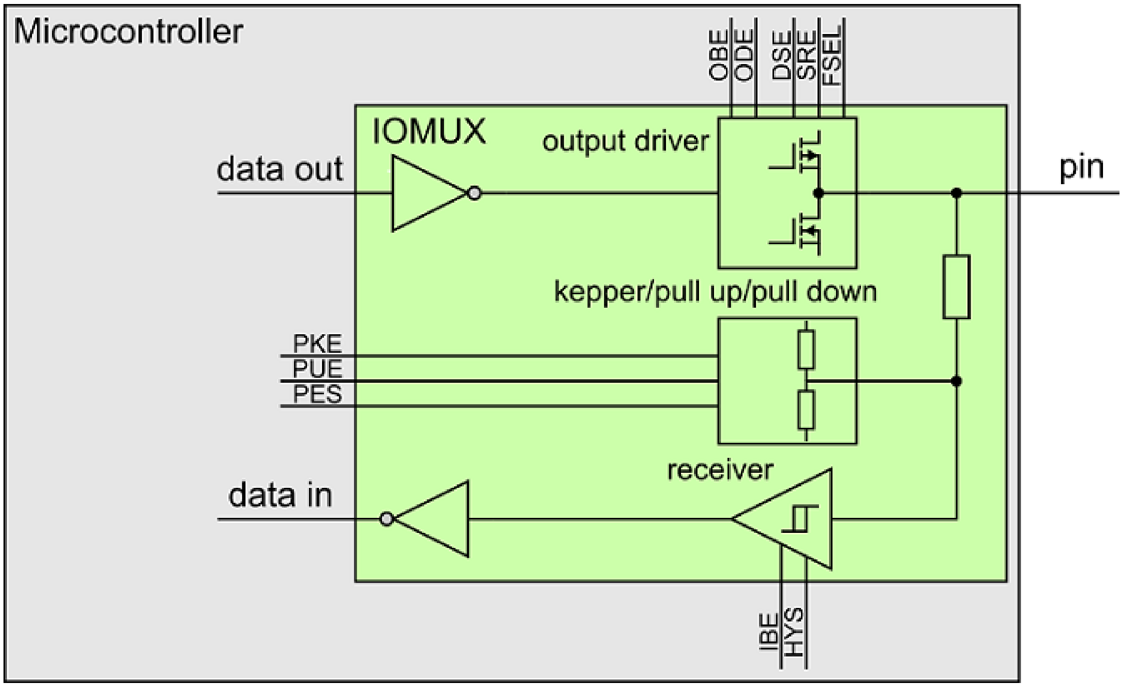


Figure 6. Simplified internal structure of the IOMUX for each pin

Pins have a three-state logic. If no device is enabled, then the pins are in high impedance state. It means that the output driver is not controlling the state of a connected circuit.

4 Input

An input can be set by two bits – IBE and HYS. The captured input signal value on a pin is loaded into the appropriate *data-in* register.

4.1 IBE

The input buffer enable (IBE) enables the input receiver. It connects a pin to the appropriate *data-in* register, which can be read by the core or any other master on the internal bus of an MCU.

Table 3. Parameters of the input set as CMOS

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit	Notes
Vih	High-level DC input voltage	–	0.7*ovdd	–	ovdd	V	1
Vil	Low-level DC input voltage	–	0	–	0.3*ovdd		

The voltage level of an input signal is set as the CMOS standard by default. The voltage levels are shown in Table 3. The logic 1 and logic 0 ranges are defined. The input signal voltage level should not be between those level areas, because the captured logic value cannot be guaranteed.

This option is used for connecting outputs of a digital component (like a memory or a transceiver) to the MCU input.

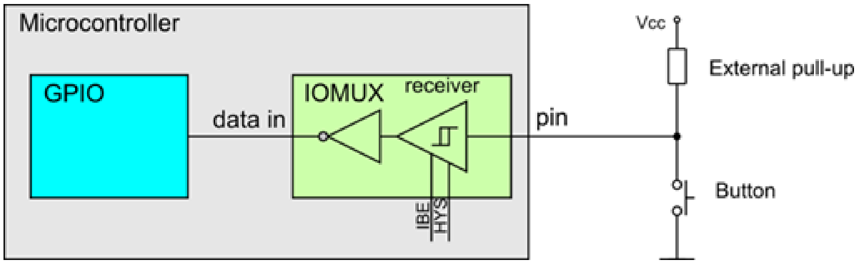


Figure 7. Connecting a button to an MCU

The simplest input usage is to connect a push-button to an MCU, as shown in Figure 7. Please note that the push-button cannot operate without a pull-up defining the default state when the push-button is not pressed.

4.2 HYS

The hysteresis (HYS) bit controls whether a pin acts as a Schmitt trigger, or a comparator remembering its last input state (hysteresis). An appropriate bit in the *data-in* register retains its value until the input level on the pin changes sufficiently to trigger the change.

Table 4. Parameters of an input with hysteresis

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit	Notes
Vhys	Input Hysteresis	ovdd = 1.8 V or 3.3 V	250	–	–	mV	–
Vt+	Schmitt trigger VT+	–	0.5*ovdd	–	–	V	1 , 2
Vt–	Schmitt trigger VT–	–	–	–	0.5*ovdd		

The voltage levels are based on half of the power supply – see Table 4. The transition from logic 0 to logic 1 occurs only when the input signal is higher than half of the power supply voltage plus half of the hysteresis and vice versa. There is no intermediate level like on the CMOS input.

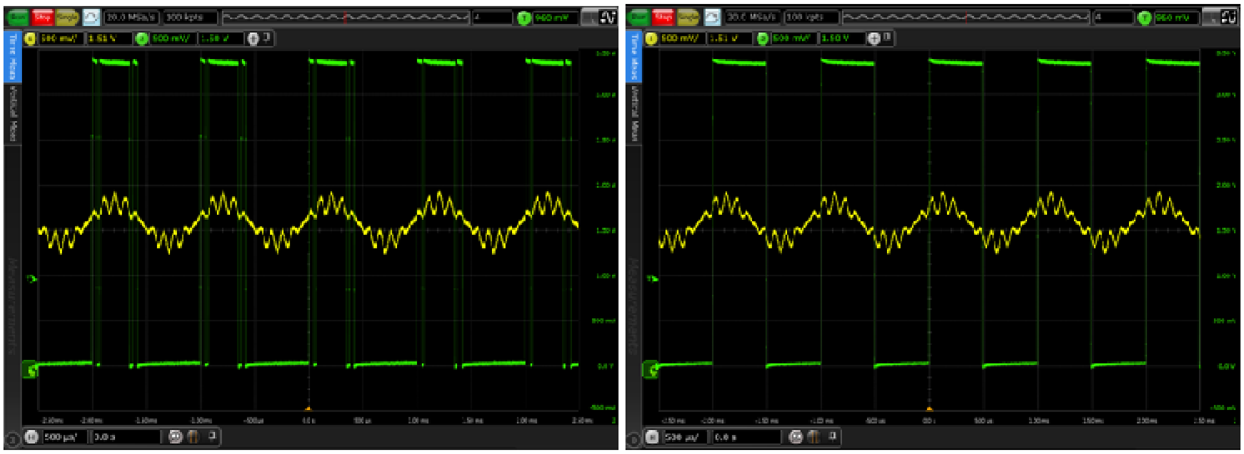


Figure 8. Input signal captured without and with hysteresis

This option is used when the input signal is noisy, to prevent glitches in the *data-in* register. Usage of the HYS option is displayed in Figure 8. The input signal (in yellow) is noisy. The green signal level is captured in the *data-in* register and sent back to a different output pin. On the left side, a normal CMOS input option is used. This causes several quickly successive changes in the captured signal safely suppressed when the HYS option is used (on the right side of the picture).

Compared to the CMOS configuration option, the HYS one slightly increases the pin power consumption as well as the propagation delay by several nanoseconds.

5 Output

The output can be set by the OBE, ODE, DSE, SRE, and SPEED bits. If enabled, a logic level from the *data-out* register appears on the appropriate output pin as an equivalent voltage level. The DSE, SRE and SPEED bits are explained in Section 7, “Electrical setting.”

5.1 OBE

The output buffer enable (OBE) enables the output driver. It connects the *data-out* register (loaded by the core) to a pin.

This option is used for connecting the inputs of the digital component (like a memory or a transceiver) to the MCU's output. Another example is the PWM signal output in motor control applications.

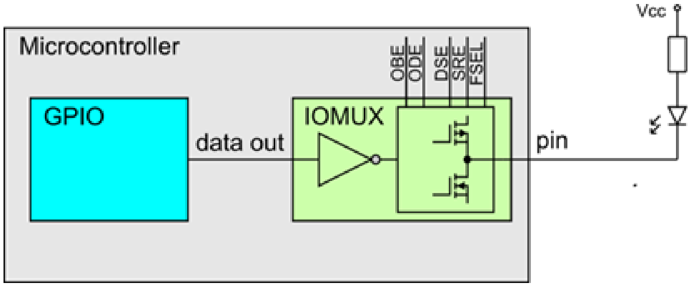


Figure 9. Connection of a LED to the MCU

The simplest output usage is to connect a LED to the MCU, as shown in Figure 9.

5.2 ODE

The open drain enable (ODE) is an option for the OBE. While the standard OBE drives logic 1 and logic 0 using internal transistors, the ODE drives only logic 0. The drain of an internal transistor is open. It means that logic 1 has to be driven by another, internal or external component.

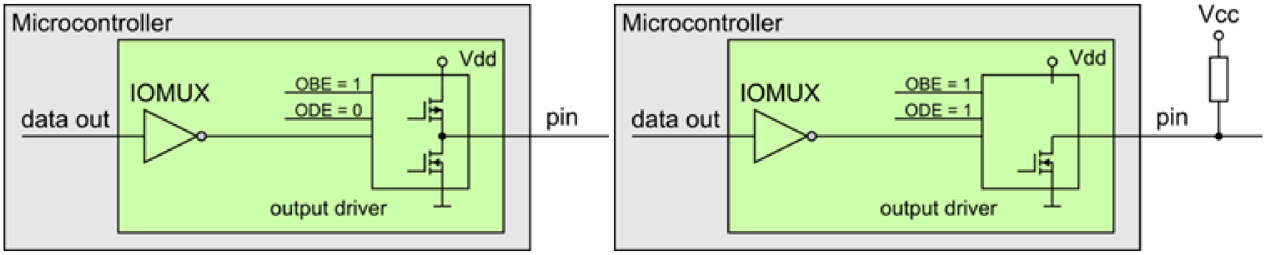


Figure 10. Difference between the OBE and ODE input drivers

Figure 10 captures the difference between the OBE and ODE modes. A fully-functional output driver is on the left side. On the right side, there is an output driver in ODE mode; the top transistor is disabled in the ODE mode.

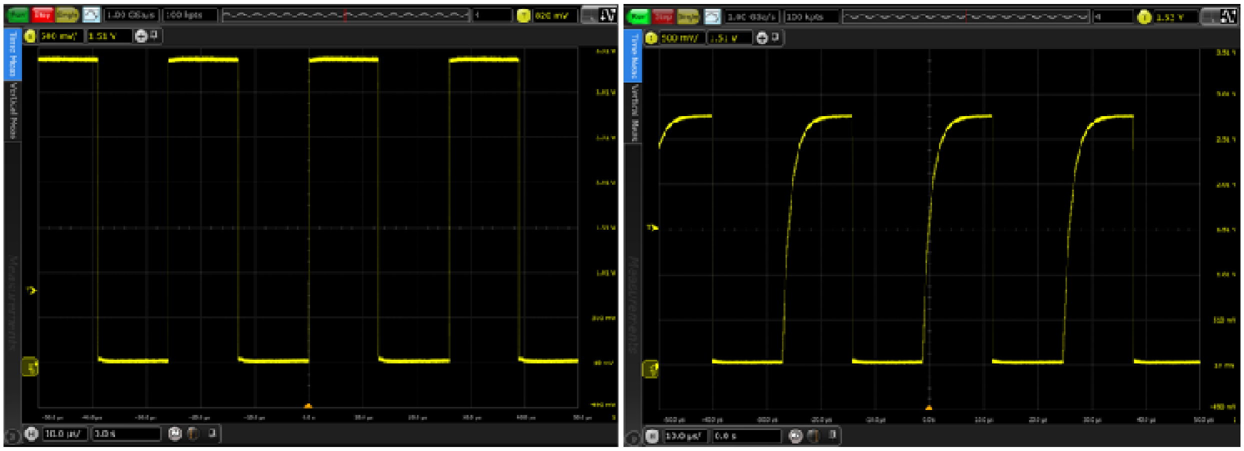


Figure 11. The OBE and ODE output driver modes

Figure 11 shows a fully functional output driver waveform on the left side. On the right side, there is a waveform of the output driver in ODE mode with an internal 22 kΩ pull-up resistor. A significant difference in the rising edge can be seen; it is due to the low-pass filter structure formed by the relatively high-value pull-up resistor and the total capacitance of the pin and the connected circuit. The falling edge is sharp thanks to the internal transistor with much lower internal resistance.

This option is essential for communication systems like One-Wire, I^2C or I^2S, where the connection between an MCU and an external component is bi-directional. In such applications, it is not possible to drive one end of the line to logic 1 and the other end to logic 0. It is called “short,” and should be avoided. Instead, there is one pull-up with a value of hundreds of ohms that defines the default state and any connected device can drive the line to logic 0.

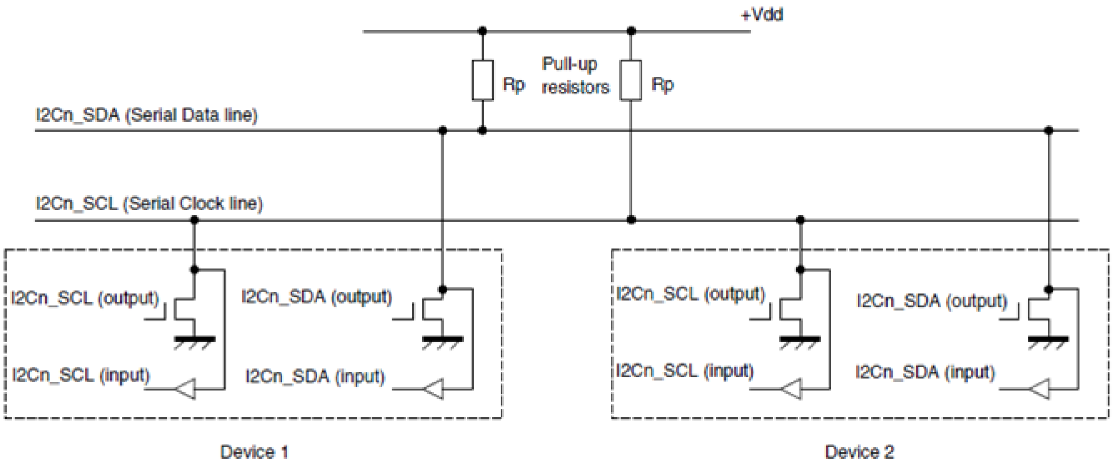


Figure 12. ODE option usage in I^2C bus

Usage of the ODE feature together with an external resistor enables communication functionality; on the other hand, it increases the total power consumption. When a node on a bus drives the line to logic 0, current flows through the pull-up resistor. For example, when a 500 Ω resistor is used, additional 6.6 mA is drawn from the 3.3 V rail.

6 Pull / keeper

The pull / keeper device is controlled by the PKE, PUE and PUS bits. The pull / keeper can be enabled by the pull / keep enable (PKE) bit. When the pull / keeper is enabled, there can be selected which part of the device is enabled by the PUE (pull-up enable) bit.

6.1 Pull-up / pull-down

The pull-ups connect a pin to the Vdd via a resistor. The pull-downs connect a pin to to the ground via a resistor. The pull-ups or pull-downs are essential in applications, where the default value of an input pin must be defined, especially in cases where the input device may be disconnected on purpose or accidentally.

When the pull-up is enabled, it is possible to select which resistance will be connected internally using the pull-up / pull-down select field (PUS). For example, the set in the i.MX / Vybrid family uses two bits and options include 100 kΩ pull-down, 22 kΩ pull-up, 47 kΩ pull-up, and 100 kΩ pull-up.

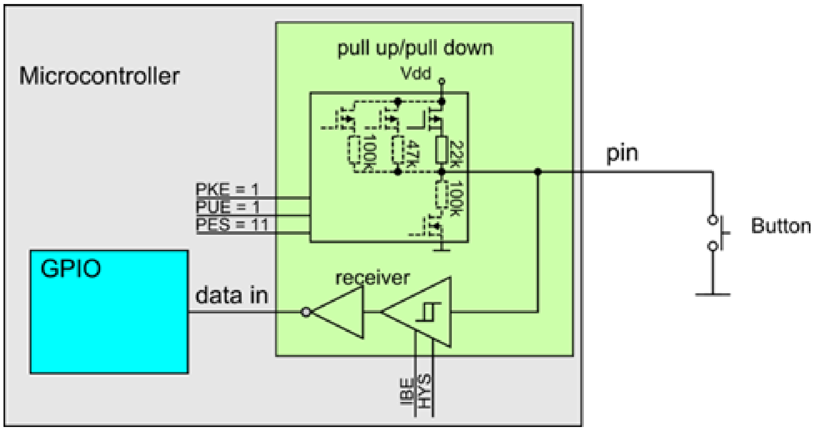


Figure 13. Internal pull-up usage

For example, it defines the pin’s input level, when the push-button is not pressed. It is a quite common application, so it makes sense to integrate it into an MCU. For example, the 22 kΩ pull-up is enabled in Figure 13. The scheme is simplified for better understanding.

Please note that an internal pull-up increases power consumption on the Vdd IO power rail, which can be significant in the Stop modes. For example, when a 22 kΩ is used it consumes up to 212 µA in the logic 0 state (according to VF6xx datasheet value).

6.2 Keeper

The keeper device is able to keep the previous output value when the output driver is disabled.

Table 5. Parameters of the keeper

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit	Notes
Rkeep	Pad keeper resistance	–	105	130	205	kΩ	4

The internal resistance of the keeper is much higher than the internal resistance of a standard output driver – see Table 5.

This option is used to save power in applications that need a lot of energy when the logic value is changed, but little energy when it is needed to just keep the state; for example, applications with a high portion of capacity part of the load. Another type of application is when it must be ensured that the pin is driven in any case, even with much less strength, so the output pin does not float and its level is defined.

6.3 SION

The software input on (SION) bit is an option to force an input path to be active regardless of the value driven by the corresponding module. It is used when the nature direction of a pin depending on selected alternative function is an output, but it is needed to read the real logic value on a pin. An example is shown in Section 8.2, "Setting the direction by selected alternative function module."

The SION bit can be used in:

- Loopback: the module of a selected alternative function drives the pad and also receives the pad value as an input
- GPIO capture: the module of a selected alternative function drives the pin and the value is captured by the GPIO

7 Electrical setting

The electrical setting on each pin simply sets how much current the IO pin will source / sink. This setting has a high influence on the current and, consequently, on the EMC noise caused by the pin. Together with the load capacitance, it sets the maximal slope of rising and falling edges.

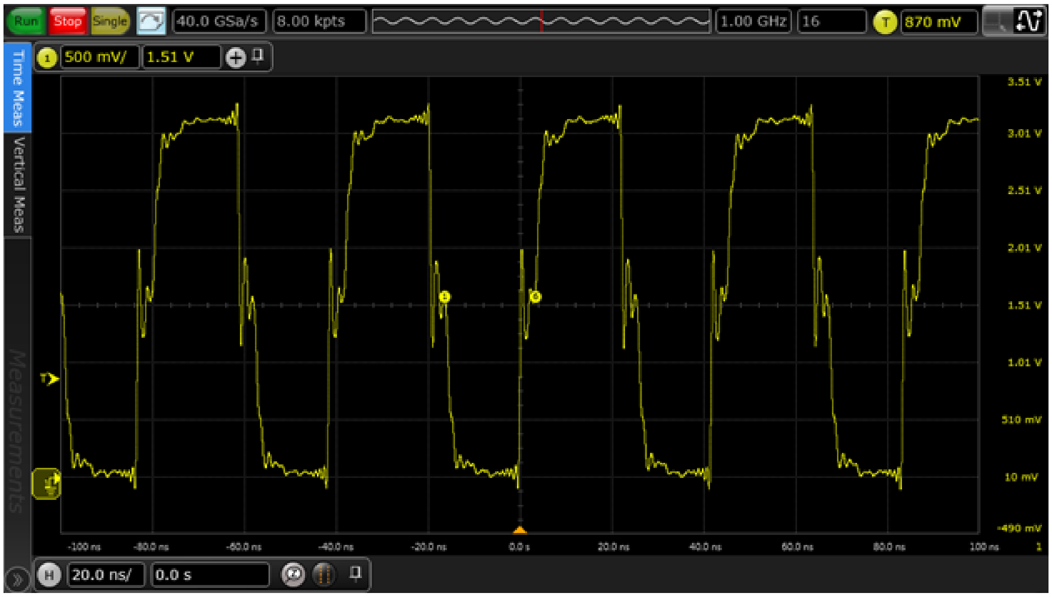


Figure 14. A pin set incorrectly for 24 MHz signal (DSE = 50 Ω SRE = high, SPEED = 200 MHz)

The pin settings have to be adjusted according to the application. It is not appropriate to set all the parameters for maximum performance. Sourcing too much current to a pin can result in spikes in the output signal which may generate noise at high frequencies. This may cause EMC compatibility issues, or end application stability issues. The incorrect pin setting example is shown in Figure 14.

7.1 DSE

The drive strength enable (DSE) can be explained as series resistance between an ideal driver’s output and its load. To achieve maximal transferred power, the impedance of the driver has to match the load impedance.

Table 6. Setting of the output driver impedance

Symbol	Parameter	Drive strength	Min	Typ	Max	Unit
Rdrv	Output driver impedance	0 0 1	116	150	220	Ω
		0 1 0	58	75	110	
		0 1 1	39	50	73	
		1 0 0	30	37	58	
		1 0 1	24	30	46	
		1 1 0 Extra drive strength	20	25	38	
		1 1 1	17	20	32	

Table 6 shows the output driver impedance of a pin depending on the DSE bit field setting.

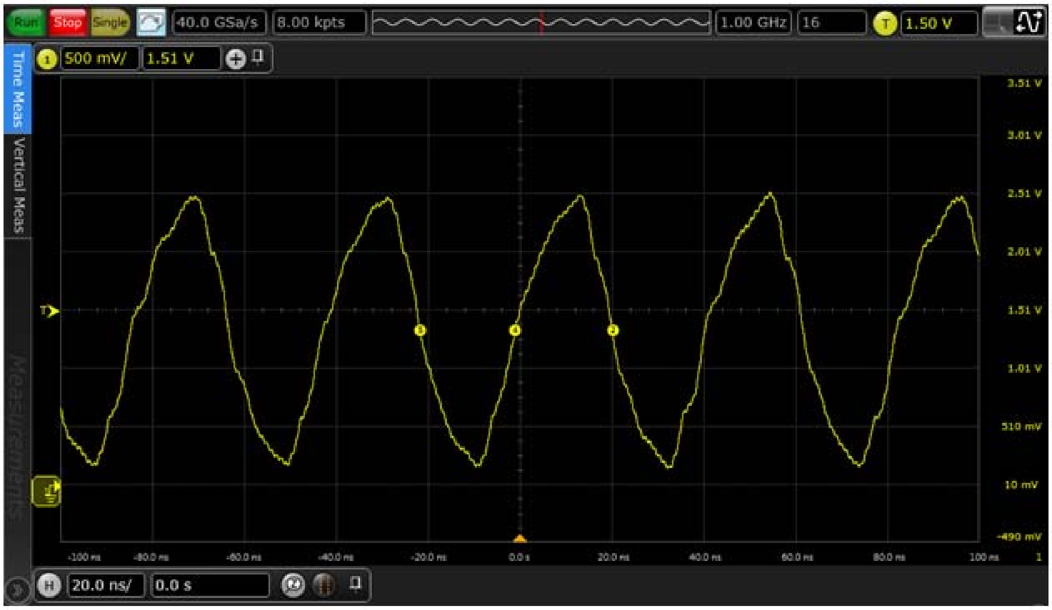


Figure 15. Incorrectly set DSE for a 24 Hz signal (DSE = 150 Ω SRE = low, SPEED = 50 MHz)

Figure 15 shows an incorrectly set DSE. The output signal voltage range is insufficient. The edge slopes are low. The equivalent series impedance of the pin is too high for the total load capacitance. The circuit behaves like an low-pass RC filter.

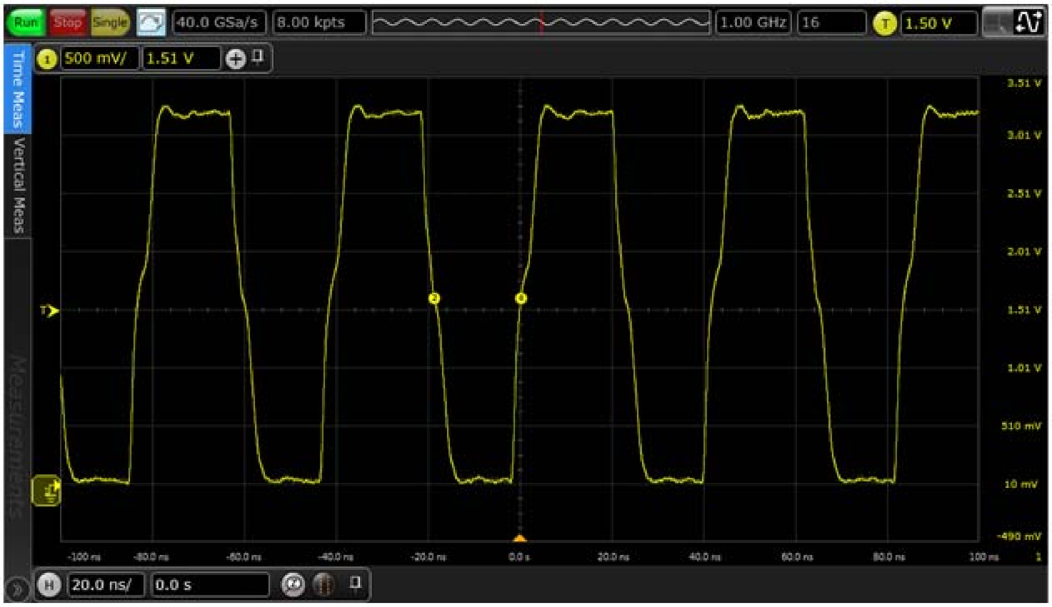


Figure 16. Correctly set DSE for 24 Hz signal (DSE = 37 Ω SRE = low, SPEED = 50 MHz)

The correct setting for the same case is shown in Figure 16. The pin’s drive strength is increased. The signal voltage levels for both logic values are correct and edge slopes sufficient. There are no high frequency spikes.

7.2 Slew rate (SRE)

The slew rate enable (SRE) bit controls how fast the pin toggles between the two logic states. Since rapidly changing states consume more power and generate spikes, it should be enabled only when necessary. Low slew rates are preferred, except for the quick control signals like parallel interfaces: EIM, EBI, SPI, or SDRAM, which need fast toggling. The incorrect setting is presented in Figure 14.

Table 7 shows the slew rate and output impedance influence on the transition times and propagation delays of the pin path. The data can be found in the datasheet.

Table 7. Influence of slew rate and output impedance

Symbol	Parameter	Drive strength	Slew rate	Test conditions	Min	Max	Unit
tpr	IO Output Transition Times (PA1), rise / fall	Max 1 1 1	slow	15 pF Cload on pad, input edge rate 200 ps	1.70	1.81	ns
			fast		1.04	1.18	
		High 1 0 1	slow		2.30	2.44	
			fast		1.69	1.79	
		Medium 1 0 0	slow		3.07	3.31	
			fast		2.45	2.61	
		Low 0 1 1	slow		5.13	5.44	
			fast		4.79	5.18	
tpo	IO Output Propagation Delay (PA2), rise / fall	Max 1 1 1	slow	15 pF Cload on pad, input edge rate 200 ps	5.01	5.04	ns
			fast		3.06	3.10	
		High 1 0 1	slow		5.55	5.68	
			fast		3.52	3.55	
		Medium 1 0 0	slow		6.37	6.67	
			fast		4.04	4.11	
		Low 0 1 1	slow		7.39	7.60	
			fast		5.54	6.10	
tpv	Output Enable to Output Valid Delay, rise / fall	Max 1 1 1	slow	15 pF Cload on pad, input edge rate 200 ps, 0->1, 1->0 pad transitions	5.12	5.21	ns
			fast		3.18	3.28	
		High 1 0 1	slow		5.72	5.80	
			fast		3.67	3.71	
		Medium 1 0 0	slow		6.55	6.80	
			fast		4.06	4.09	
		Low 0 1 1	slow		7.80	8.19	
			fast		5.72	6.22	
tpl	Input Pad Propagation Delay rise / fall	without hysteresis	–	150 F Cload on, input edge rate from pad = 1.2 ns	1.06	1.31	ns
		with hysteresis	–		1.22	1.41	

7.3 SPEED

The SPEED is a selectable bit field that sets electrical characteristics of a pin in a given frequency range. There are additional 2-bit slew rate control signals. For the Vybrid and i.MX families, the frequency range may be selected between 50, 100 and 200 MHz. These options can either increase the output driver current in the higher frequency range, or reduce the switching noise in the lower frequency range.

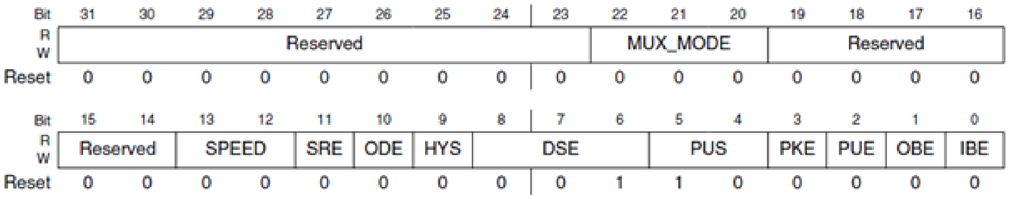
8 Pin direction design specific setting

There are different ways of setting the direction of a pin in the IOMUX on Freescale devices:

- Setting the direction of a pin in the IOMUX module using the IBE and OBE bits. This option is used on the Vybrid devices. A similar setting is used on the Power architecture devices.
- Setting the pin direction using a selected alternative function module. Setting the IBE and OBE bits automatically, based on the alternative function selected. This option is used on the i.MX6 devices. A similar setting is used on the Kinetis devices.

8.1 Setting the direction by the IOMUXC module

Each pin is controlled by the software MUX pad control register.



Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Figure 17. Software MUX pad control register

An example of a pad control register is shown in Figure 17.

8.1.1 GPIO usage setting

The pin direction is controlled manually by the IBE and OBE bits. Each pin may be set as input-only or output-only or as bi-directional.

The GPIO module contains the following registers:

- Port data input register (GPIO_PDIR)
- Port data output register (GPIO_PDOR)

When the IBE is set, then the pin input signal value is stored in the appropriate GPIO_PDIR[x].

When the OBE is set, then the requested signal logic value in the GPIO_PDOR[x] is updated to the appropriate pin output signal level. When the IBE is set, then the real pin signal value can be read from the appropriate GPIO_PDIR[x].

The pin output can also be set using the GPIO_PSOR[x], cleared using the GPIO_PCOR[x] or toggled using the GPIO_PTOR[x].

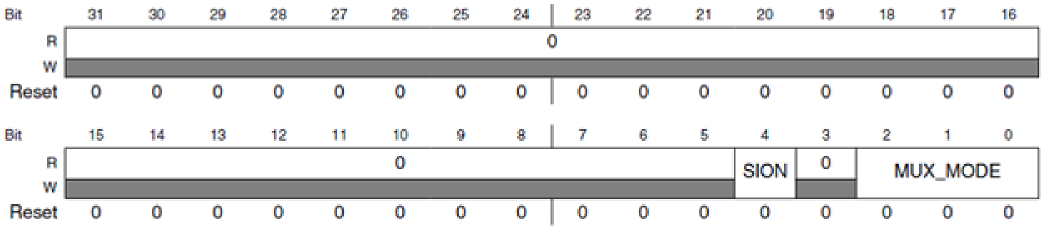
8.1.2 I^2C / I^2S usage setting

The I^2C / I^2S require bi-directional communication. According to this design, the following must be enabled in the IOMUX:

- Input path by IBE = 0b1
- Output path by OBE = 0b1
- Open drain function by ODE = 0b1
- Internal pull-up for very low clock frequencies PKE = 0b1, PUE = 0b1, PUS = 0b11 or external pull-up for regular or high frequencies

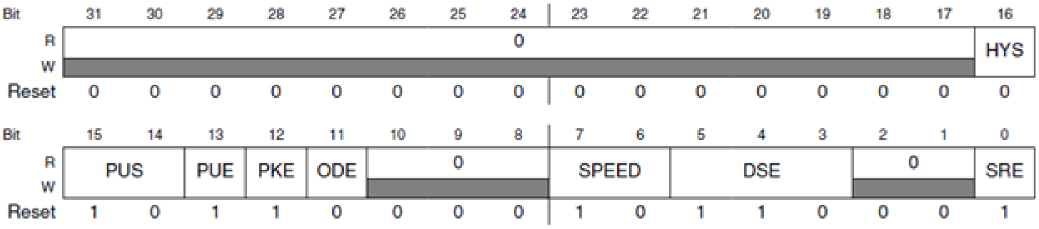
8.2 Setting the direction by selected alternative function module

Each pin is controlled by two registers: pad MUX register and pad control register.



Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 18. Pad MUX register



Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

Figure 19. Pad control register

Examples of the pad control registers are shown in Figure 18 and in Figure 19.

8.2.1 GPIO usage setting

The GPIO module contains the following registers:

- Data register (GPIO_DR)
- GPIO direction register (GPIO_GDIR); GPIO bit direction
- Pad sample register (GPIO_PSR)

Each GPIO register controls 32 pins [0:31]. Setting the pin direction in the GPIO register automatically changes the IBE and OBE bit settings in the IOMUX module.

When the GPIO pin is configured as an input (GPIO_GDIR[x] = 0), then the pin input signal value is stored in the appropriate GPIO_DR[x].

When the GPIO pin is configured as an output (GPIO_GDIR[x] = 1), then the requested signal logic value in the GPIO_DR[x] is updated to the appropriate pin output signal level. In this case, it is not possible to read the real signal level on the pin using the GPIO_DR[x]. To read the real signal level, the SION[x] bit has to be used to enable the input path. The real signal value is then stored in the read-only register GPIO_PSR[x].

8.2.2 I^{2}C / I^{2}S usage setting

The I^{2}C / I^{2}S require bi-directional communication. According to this design, the following must be enabled in the IOMUX:

- Output path by OBE = 0b1 (enabled automatically by selecting the I^{2}C alternate function of the pin)
- Open drain function by ODE = 0b1
- Input path by SION = 0b1
- Internal pull-up for very low clock frequencies PKE = 0b1, PUE = 0b1, PUS = 0b11, or external pull-up for normal and high frequencies

9 Conclusion

Each pin’s operation depends on configuration of the appropriate IOMUX controller's register. Each register consists of a combinatorial logic built of several basic IOMUX cells. Each IOMUX cell handles only one pin signal muxing. With that being said, the pin settings have significant influence on the application behavior and performance. The effort put on pins setting should be of the same order of magnitude as that put of the code development.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number: AN5078
Rev. 0
02/2015