XMC1000

Microcontroller Series
for Industrial Applications

PORTS

✓ Digital Port Logic

Device Guide

V1.0 2013-05

Microcontrollers

Edition 2013-05

Published by
Infineon Technologies AG
81726 Munich, Germany

© 2013 Infineon Technologies AG
All Rights Reserved.

Legal Disclaimer

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Revision History	
Page or Item	**Subjects (major changes since previous revision)**
V1.0, 2013-05	

Trademarks of Infineon Technologies AG

AURIX™, C166™, CanPAK™, CIPOS™, CIPURSE™, EconoPACK™, CoolMOS™, CoolSET™, CORECONTROL™, CROSSAVE™, DAVE™, EasyPIM™, EconoBRIDGE™, EconoDUAL™, EconoPIM™, EiceDRIVER™, eupec™, FCOS™, HITFET™, HybridPACK™, I²RF™, ISOFACE™, IsoPACK™, MIPAQ™, ModSTACK™, my-d™, NovalithIC™, OptiMOS™, ORIGA™, PRIMARION™, PrimePACK™, PrimeSTACK™, PRO-SIL™, PROFET™, RASIC™, ReverSave™, SatRIC™, SIEGET™, SINDRION™, SIPMOS™, SmartLEWIS™, SOLID FLASH™, TEMPFET™, thinQ!™, TRENCHSTOP™, TriCore™.

Other Trademarks

Advance Design System™ (ADS) of Agilent Technologies, AMBA™, ARM™, MULTI-ICE™, KEIL™, PRIMECELL™, REALVIEW™, THUMB™, µVision™ of ARM Limited, UK. AUTOSAR™ is licensed by AUTOSAR development partnership. Bluetooth™ of Bluetooth SIG Inc. CAT-iq™ of DECT Forum. COLOSSUS™, FirstGPS™ of Trimble Navigation Ltd. EMV™ of EMVCo, LLC (Visa Holdings Inc.). EPCOS™ of Epcos AG. FLEXGO™ of Microsoft Corporation. FlexRay™ is licensed by FlexRay Consortium. HYPERTERMINAL™ of Hilgraeve Incorporated. IEC™ of Commission Electrotechnique Internationale. IrDA™ of Infrared Data Association Corporation. ISO™ of INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. MATLAB™ of MathWorks, Inc. MAXIM™ of Maxim Integrated Products, Inc. MICROTEC™, NUCLEUS™ of Mentor Graphics Corporation. Mifare™ of NXP. MIPI™ of MIPI Alliance, Inc. MIPS™ of MIPS Technologies, Inc., USA. muRata™ of MURATA MANUFACTURING CO., MICROWAVE OFFICE™ (MWO) of Applied Wave Research Inc., OmniVision™ of OmniVision Technologies, Inc. Openwave™ Openwave Systems Inc. RED HAT™ Red Hat, Inc. RFMD™ RF Micro Devices, Inc. SIRIUS™ of Sirius Satellite Radio Inc. SOLARIS™ of Sun Microsystems, Inc. SPANSION™ of Spansion LLC Ltd. Symbian™ of Symbian Software Limited. TAIYO YUDEN™ of Taiyo Yuden Co. TEAKLITE™ of CEVA, Inc. TEKTRONIX™ of Tektronix Inc. TOKO™ of TOKO KABUSHIKI KAISHA TA. UNIX™ of X/Open Company Limited. VERILOG™, PALLADIUM™ of Cadence Design Systems, Inc. VLYNQ™ of Texas Instruments Incorporated. VXWORKS™, WIND RIVER™ of WIND RIVER SYSTEMS, INC. ZETEX™ of Diodes Zetex Limited.

Last Trademarks Update 2011-02-24

Table of Contents

Table of Contents **4**

Digital Port Logic **5**

1 General Purpose I/O Port **6**

2 Input **7**
2.1 Input Pin Setup 7
2.2 Available Input Pins 7
2.3 Pad Hysteresis 7
2.3.1 Pad Hysteresis Setup 8
2.3.2 Pins Available for Pad Hysteresis 8
2.4 Internal Pull Devices 8

3 Output **9**
3.1 Output Pin Setup 9
3.2 Bit-banding support in XMC1000 9
3.3 Available Output Pins 9
3.4 Output Driver Mode 10

4 Alternate Output **11**
4.1 Set up Alternate Output Pin 11
4.2 Available Alternate Output Pins 11

5 Hardware Controlled I/O **12**
5.1 Hardware Control Setup 12
5.2 Hardware Pins for Peripherals 12

6 Power Save Mode **14**
6.1 Configure Pin in Power Save Mode 14

7 Ports Configuration during Power-up until end of Startup Software execution **15**
7.1 User Productive mode 15
7.2 SSC BSL mode 17
7.3 Debug mode (SWD) 18
7.4 Debug mode (SPD) 19
7.5 ASC BSL half-duplex mode 20
7.6 ASC BSL full-duplex mode 21

Digital Port Logic

1 General Purpose I/O Port

The basic feature of the Ports is the General Purpose Input and Output functionality. The following sections describe how to configure the pins for input, output and alternate output functions.

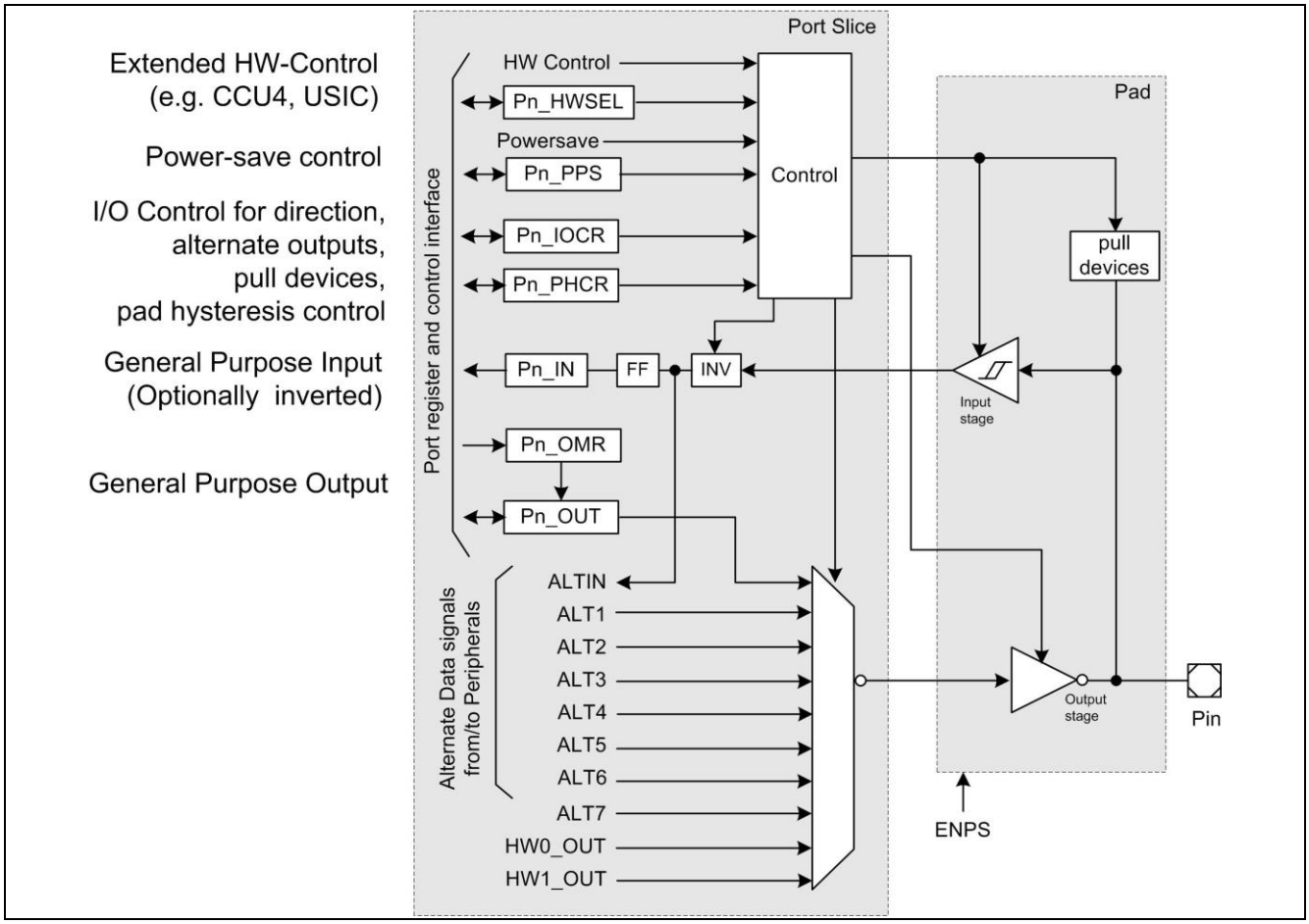


Figure 1 Block Diagram of a Digital Port Pin

2 Input

2.1 Input Pin Setup

The following C code snippet shows how to configure a pin as an input.

```
// Set Port P0.0 as input
PORT0->IOCR0 &= ~PORT0_IOCR0_PC0_Msk;
PORT0->IOCR0 |= 0x0 << PORT0_IOCR0_PC0_Pos;
P0_0_read = PORT0->IN & PORT0_IN_P0_Msk; // Returns input level
```

Note: A Ports header file (GPIO.h) is available in the DAVE Easy Start project.

2.2 Available Input Pins

Table 1 shows the available pins in the different packages of the XMC1xxx, configurable as an input.

Table 1 Input pins available for TSSOP-38, 28 and 16

Port	TSSOP-38	TSSOP-28	TSSOP-16
P0	P0.0 - P0.15	P0.0, P0.4 - P0.10, P0.12 - P0.15	P0.0, P0.5 - P0.9, P0.14 - P0.15
P1	P1.0 - P1.5	P1.0 - P1.3	-
P2	P2.0 - P2.11	P2.0 - P2.2, P2.5 - P2.11	P2.0, P2.6 - P2.11

Note: P2.2 – P2.9 are analog inputs.

2.3 Pad Hysteresis

Figure 2 shows the standard hysteresis and large hysteresis characteristics with respect to logic and voltage.

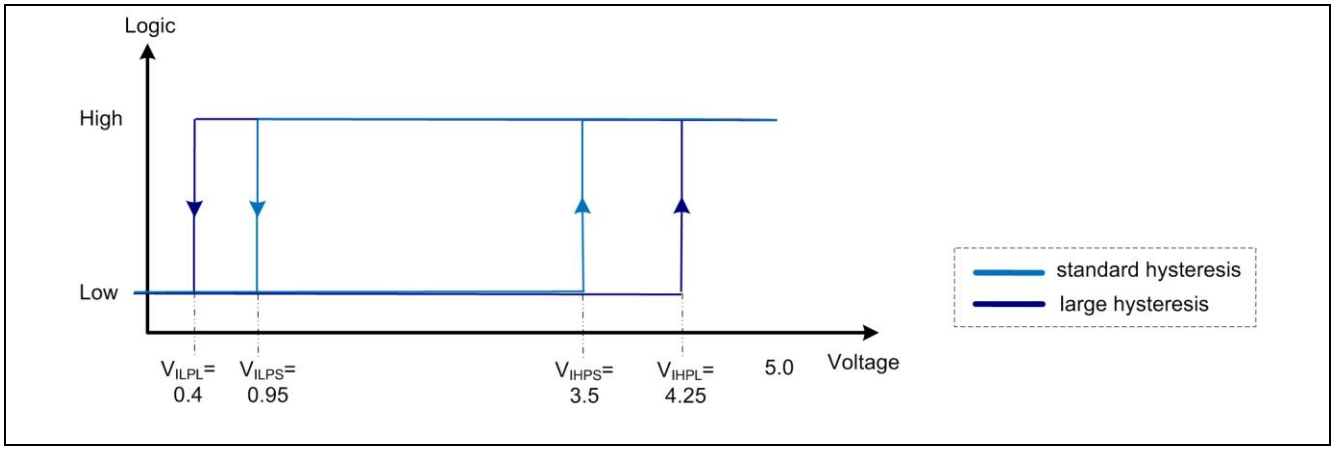


Figure 2 Graphical representation of Standard and Large Hysteresis characteristics

Pad hysteresis can be configured to standard or large hysteresis based on application needs, for example in a touch-sensing application.

For detailed example on pad hysteresis, please refer to Scheme A and Scheme B in the LED and Touch-Sense chapter of the XMC1xxx Reference Manual.

2.3.1 Pad Hysteresis Setup

The following C code snippets demonstrate how to configure the pad to standard and large hysteresis.

Note: A Ports header file (GPIO.h) is available in the DAVE Easy Start project.

```
// Standard hysteresis for Port P0.1
PORT0->PHCR0 &= ~PORT0_PHCR0_PH1_Msk;

// Large hysteresis for Port P0.1
PORT0->PHCR0 |= PORT0_PHCR0_PH1_Msk;
```

2.3.2 Pins Available for Pad Hysteresis

Pad hysteresis control is available for all of the pins in the XMC1xxx package.

2.4 Internal Pull Devices

The following C code snippets demonstrate how to configure the internal pull devices.

Note: A Ports header file (GPIO.h) is available in the DAVE Easy Start project.

```
// Internal pull-down device active (Direct Input) for Port P0.1
PORT0->IOCR0 &= ~PORT0_IOCR0_PC1_Msk;
PORT0->IOCR0 |= 0x1 << PORT0_IOCR0_PC1_Pos;

// Internal pull-up device active (Direct Input) for Port P0.1
PORT0->IOCR0 &= ~PORT0_IOCR0_PC1_Msk;
PORT0->IOCR0 |= 0x2 << PORT0_IOCR0_PC1_Pos;
```

3 Output

3.1 Output Pin Setup

The following C code snippet shows how to configure a pin as an output.

Note: A Ports header file (GPIO.h) is available in the DAVE Easy Start project.

```
// Set Port P0.0 as output
PORT0->IOCR0 &= ~PORT0_IOCR0_PC0_Msk;
PORT0->IOCR0 |= 0x10 << PORT0_IOCR0_PC0_Pos; // Set 1XXXX_B for output
```

3.2 Bit-banding support in XMC1000

XMC1000 runs on an ARM Cortex-M0 processor that does not support bit-banding. However, direct bit manipulation of the bit-banding operation can be achieved using the Output Modification Register Pn_OMR, which supports:

- set
- clear
- toggle single or multiple Port pins in a single access

The following C code snippets demonstrate how to program a single Port pin using the output modification register.

Note: A Ports header file (GPIO.h) is available in the DAVE Easy Start project.

```
// Set Port P0.0 in output register
PORT0->OMR = PORT0_OMR_PS0_Msk;

// Reset Port P0.0 in output register
PORT0->OMR = PORT0_OMR_PR0_Msk;

// Toggle Port P0.0 in output register
PORT0->OMR = PORT0_OMR_PS0_Msk | PORT0_OMR_PR0_Msk;
```

3.3 Available Output Pins

Table 2 shows the available pins in the different packages of the XMC1xxx that are configurable as an output.

Table 2 Output pins available for TSSOP-38, 28 and 16

Port	TSSOP-38	TSSOP-28	TSSOP-16
P0	P0.0 - P0.15	P0.0, P0.4 - P0.10, P0.12 - P0.15	P0.0, P0.5 - P0.9, P0.14 - P0.15
P1	P1.0 - P1.5	P1.0 - P1.3	-
P2	P2.0 - P2.1, P2.10 - P2.11	P2.0 - P2.1, P2.10 - P2.11	P2.0, P2.10 - P2.11

3.4 Output Driver Mode

The following C code snippets demonstrate how to configure the output driver modes.

Note: A Ports header file (GPIO.h) is available in the DAVE Easy Start project.

```
// Push-pull mode for Port P0.1
PORT0->IOCR0 &= ~PORT0_IOCR0_PC1_Msk;
PORT0->IOCR0 |= 0x10 << PORT0_IOCR0_PC1_Pos; // Set 10XXX_B for Push-pull mode

// Open-drain mode for Port P0.1
PORT0->IOCR0 &= ~PORT0_IOCR0_PC1_Msk;
PORT0->IOCR0 |= 0x18 << PORT0_IOCR0_PC1_Pos; // Set 11XXX_B for Open-drain mode
```

4 Alternate Output

4.1 Set up Alternate Output Pin

The following C code snippet shows how to configure a pin as an alternate output.

Note: A Ports header file (GPIO.h) is available in the DAVE Easy Start project.

```
// Set Port P0.0 as alternate output function 1
PORT0->IOCR0 &= ~PORT0_IOCR0_PC0_Msk;
PORT0->IOCR0 |= 0x11 << PORT0_IOCR0_PC0_Pos;
```

A detailed connectivity list of alternate functions to the Port pins is available in the ‘Port I/O Functions Table’ in the Ports Chapter of the XMC1xxx Reference Manual.

4.2 Available Alternate Output Pins

For the available pins configurable as an alternate output, please refer to **Table 2**.

5 Hardware Controlled I/O

5.1 Hardware Control Setup

The following C code snippets show how to assign hardware control of a pin to a peripheral.

Note: A Ports header file (GPIO.h) is available in the DAVE Easy Start project.

```
// Select hardware control path HW0 for Port P1.0
PORT1->HWSEL &= ~PORT1_HWSEL_HW0_Msk;
PORT1->HWSEL |= 0x1 << PORT1_HWSEL_HW0_Pos;

// Select hardware control path HW1 for Port P1.0
PORT1->HWSEL &= ~PORT1_HWSEL_HW0_Msk;
PORT1->HWSEL |= 0x2 << PORT1_HWSEL_HW0_Pos;
```

5.2 Hardware Pins for Peripherals

Hardware Pins for Universal Serial Interface Channel (USIC)

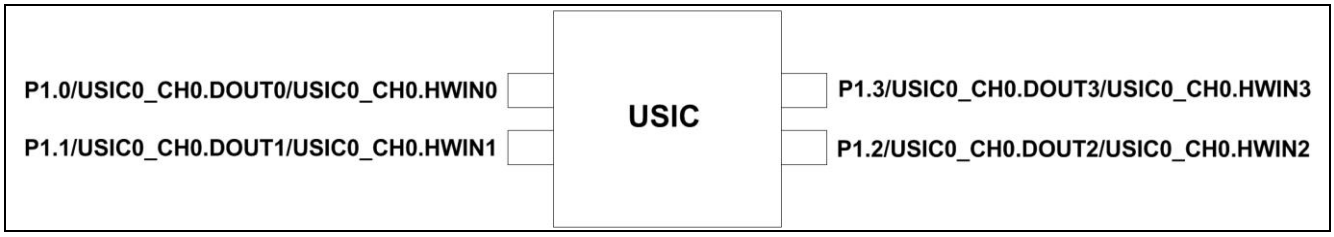


Figure 3 Hardware input and output pins for USIC

Hardware Pins for LED and Touch-Sense (LEDTS)

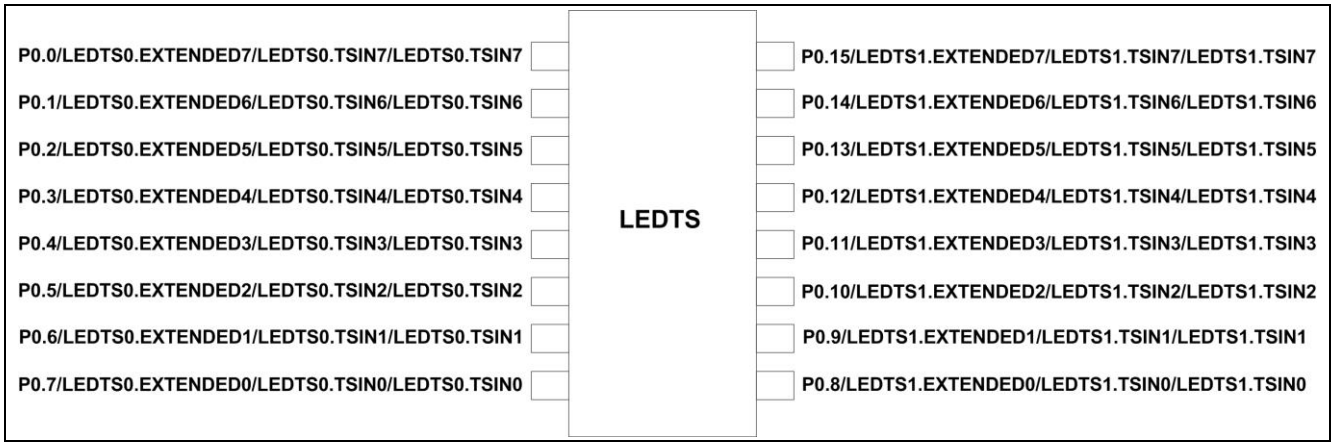


Figure 4 Hardware input and output pins for LEDTS

Hardware Pins for Capture/Compare Unit 4 (CCU4)

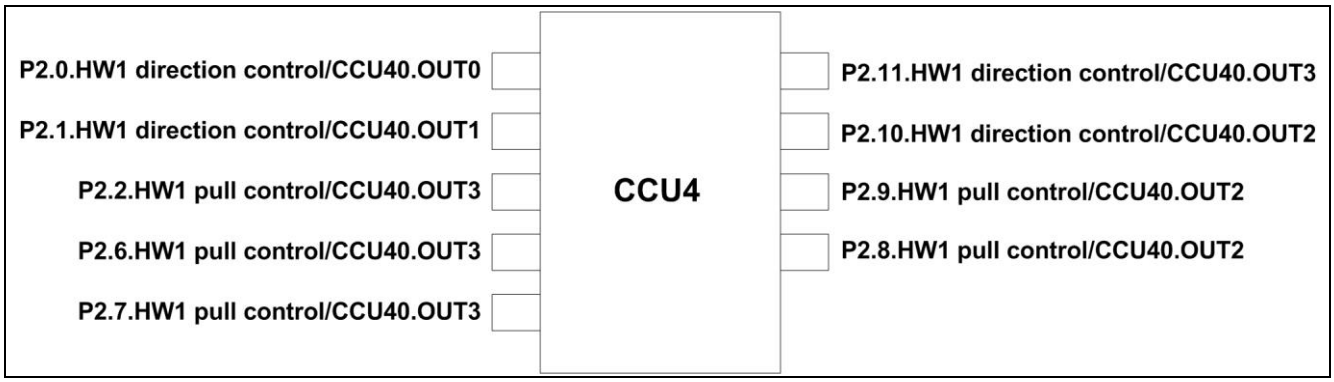


Figure 5 Hardware pins for CCU4 pull-up/down and direction control

Hardware Pins for Brightness and Color Control Unit (BCCU)

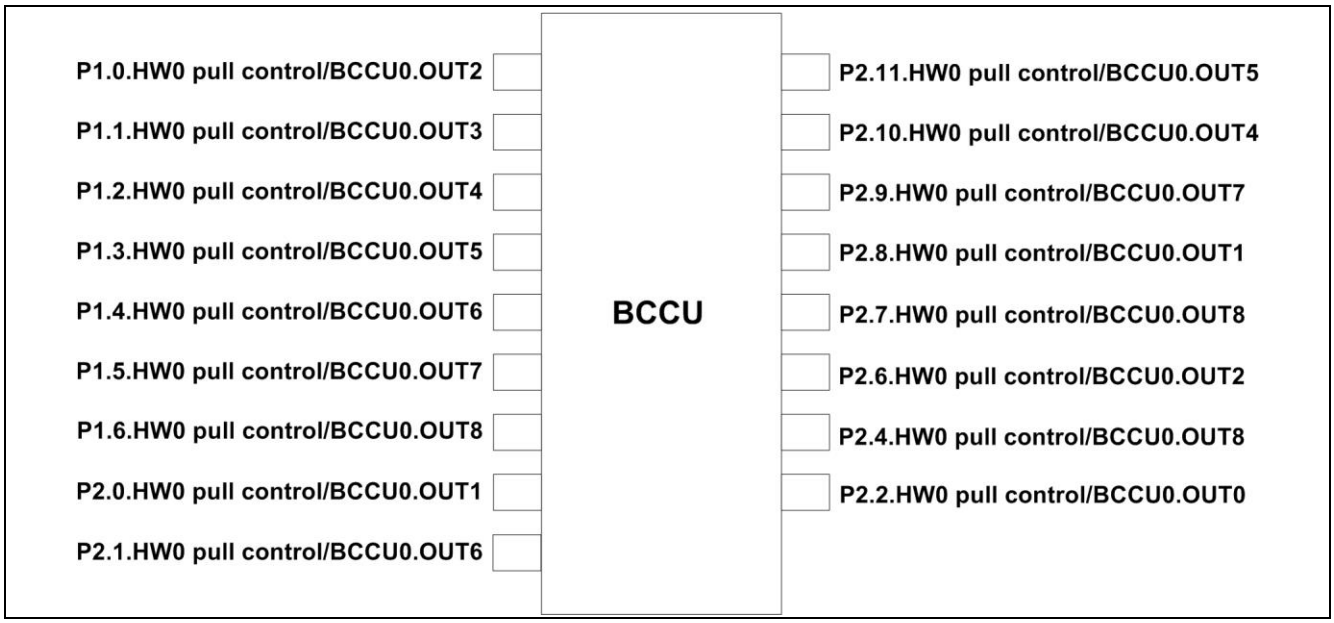


Figure 6 Hardware pins for BCCU pull-up/down control

Hardware Pins for Comparator Control Unit (CMPCU)

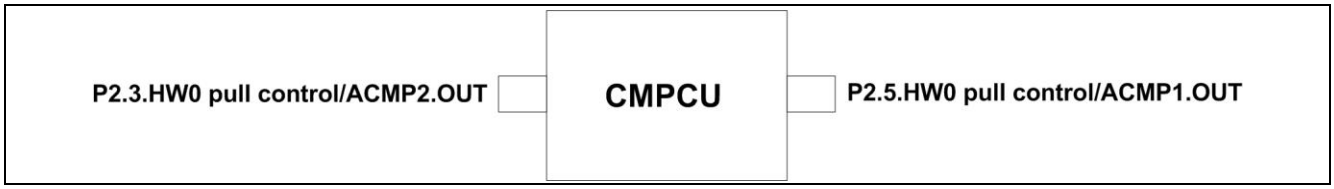


Figure 7 Hardware pins for CMPCU pull-up/down control

6 Power Save Mode

In Deep-sleep mode, when pin Power Save mode is enabled:

Output

- Output driver tri-state
- Input Pull devices off
- Input Schmitt-Trigger off

Input

- Input signal driven statically high or low, software defined by Pn_OUT

6.1 Configure Pin in Power Save Mode

The following C code snippets show how to enable and disable pin Power Save mode.
Note: A Ports header file (GPIO.h) is available in the DAVE Easy Start project.

```
// Enable pin power save for Port P0.0
PORT0->PPS |= PORT0_PPS_PPS0_Msk;

// Disable pin power save for Port P0.0
PORT0->PPS &= ~PORT0_PPS_PPS0_Msk;
```

The following C code snippets show how to configure pin power save behavior in Deep-Sleep mode.
Note: A Ports header file (GPIO.h) is available in the DAVE Easy Start project.

```
// Set pin power save behavior to Input value = Pn_OUT for Port P0.0
PORT0->IOCR0 &= ~PORT0_IOCR0_PC0_Msk;
PORT0->IOCR0 |= 0x0 << PORT0_IOCR0_PC0_Pos;

// Set pin power save behavior for output pin, Port P0.0
PORT0->IOCR0 &= ~PORT0_IOCR0_PC0_Msk;
PORT0->IOCR0 |= 0x10 << PORT0_IOCR0_PC0_Pos;
```

7 Port Configuration during Power-up until end of Startup Software (SSW) Execution

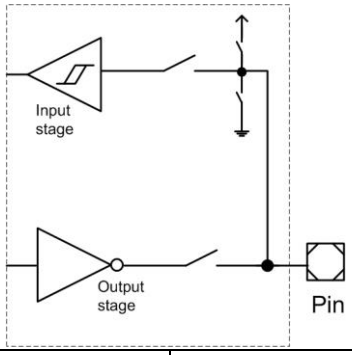
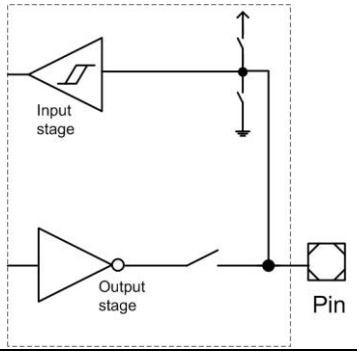
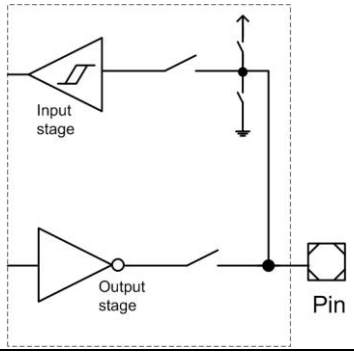
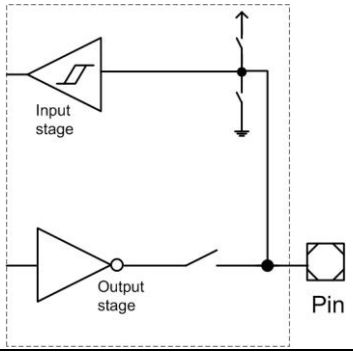
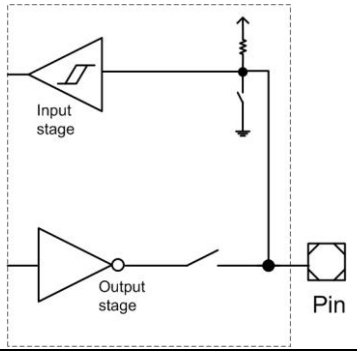
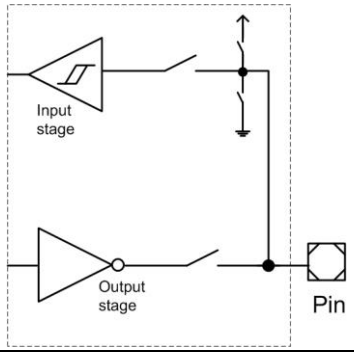
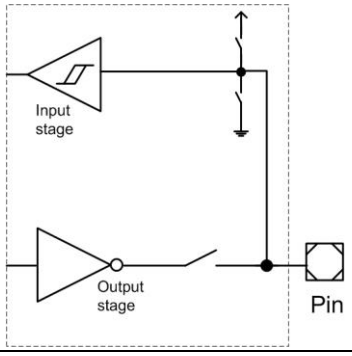
The tables that follow illustrate the Ports driver and pull configuration during power-up, after reset, and at the end of Startup Software (SSW) execution.

Note: The Port/pin configuration at the end of SSW execution will differ depending on the boot modes selected, as described in each section.

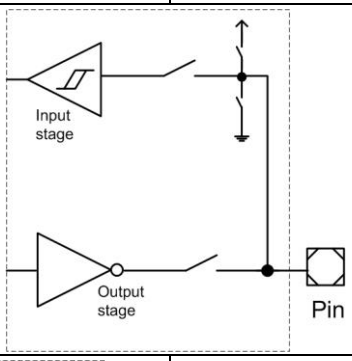
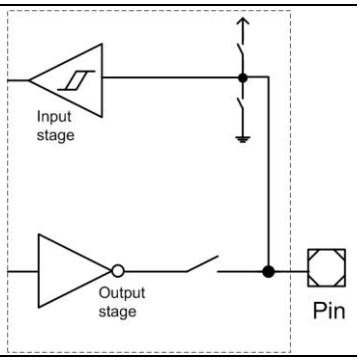
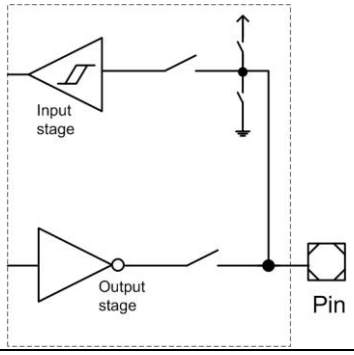
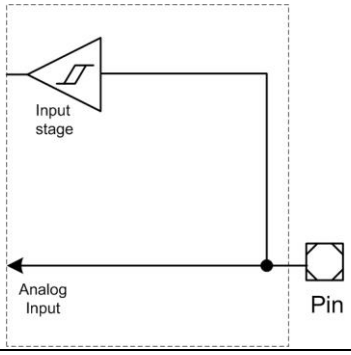
7.1 User Productive mode

Table 3 shows the Port/pin configuration for User Productive mode.

Table 3 Port/Pin Configuration for User Productive mode

Port/Pin	During Power up	After Reset (Low)	At the end of SSW execution
P0, except P0.8 and P0.15			
P0.8			
P0.15			

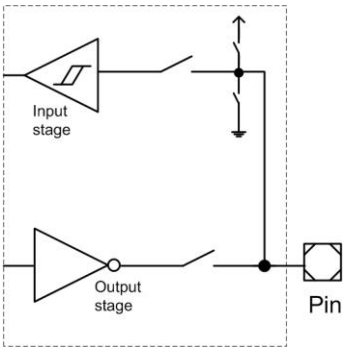
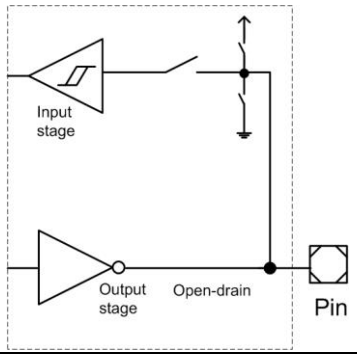
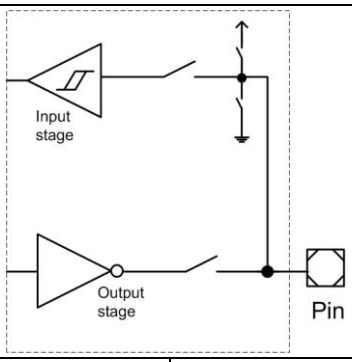
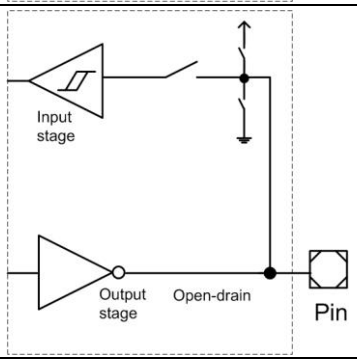
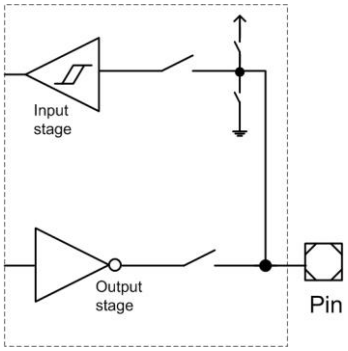
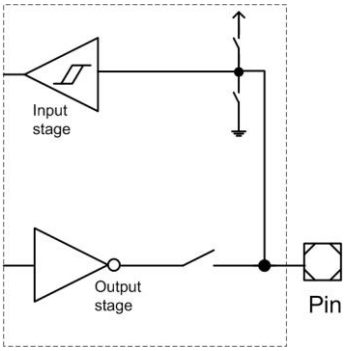
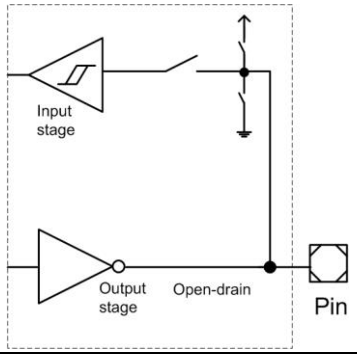
Port **Configuration during Power-up until end of Startup Software (SSW) Execution**

Port/Pin	During Power up	After Reset (Low)	At the end of SSW execution
P1			
P2			

7.2 SSC BSL mode

Table 4 shows the Port pins used for the SSC BSL mode and the corresponding pin configuration. The other Port pins not referenced in the table follow the configuration given for User Productive mode (see **Table 3**).

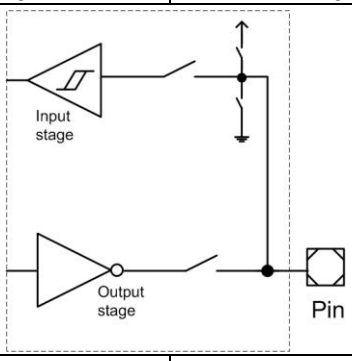
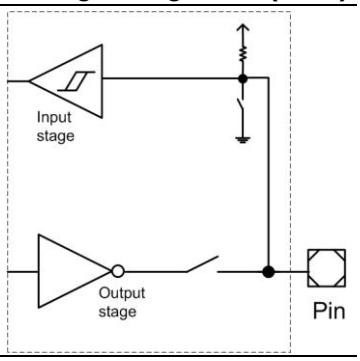
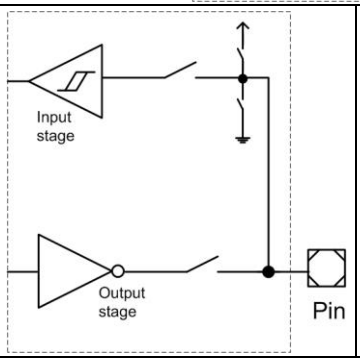
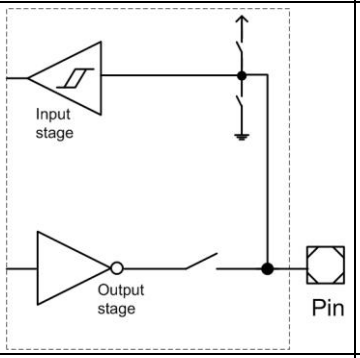
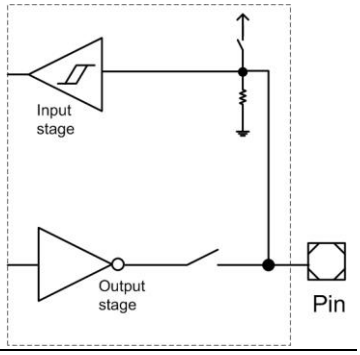
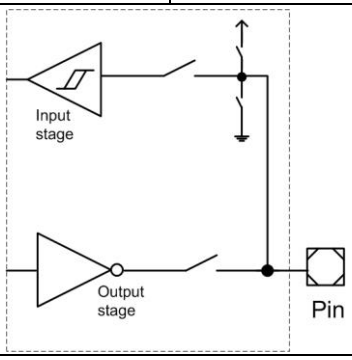
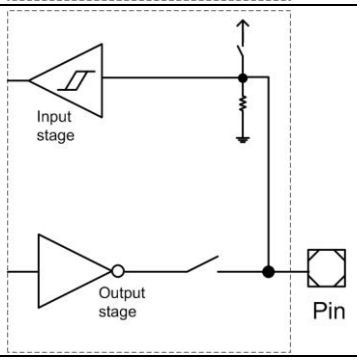
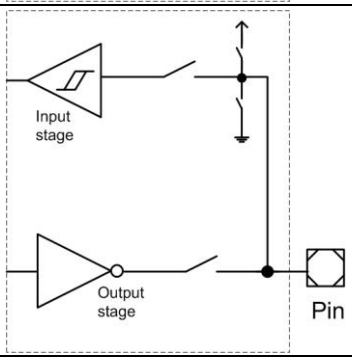
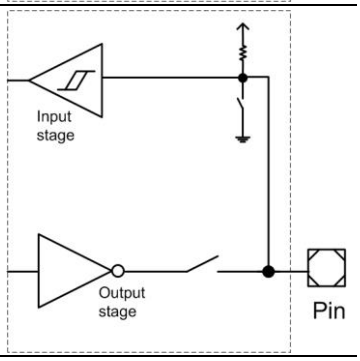
Table 4 Port/Pin Configuration for SSC BSL mode

Port/Pin	During Power up	After Reset (Low)	During SSC BSL mode
P0.13 CS(O) for SSC BSL mode			
P0.14 SCLK(O) for SSC BSL mode			
P0.15 DATA(I/O) for SSC BSL mode			

7.3 Debug mode (SWD)

Table 5 shows the Port pins used for the Debug mode (SWD) and the corresponding pin configuration. The other Port pins not referenced in the table follow the configuration given for User Productive mode (see **Table 3**).

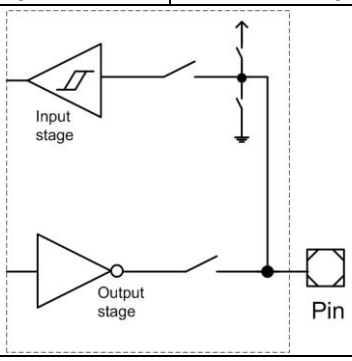
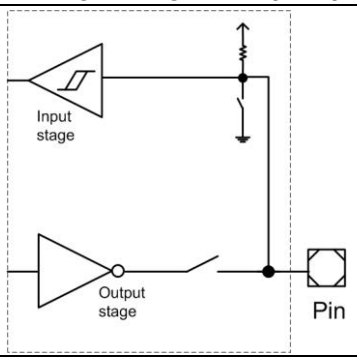
Table 5 Port/Pin Configuration for Debug mode (SWD)

Port/Pin	During Power up	After Reset (Low)	During Debug mode (SWD)
P0.14 SWDIO_0 for Debug mode (SWD)			
P0.15 SWDCLK_0 for Debug mode (SWD)			
P1.2 SWDCLK_1 for Debug mode (SWD)			
P1.3 SWDIO_1 for Debug mode (SWD)			

7.4 Debug mode (SPD)

Table 6 shows the Port pins used for the Debug mode (SPD) and the corresponding pin configuration. The other Port pins not referenced in the table follow the configuration given for User Productive mode (see **Table 3**).

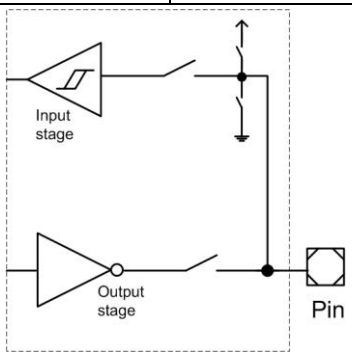
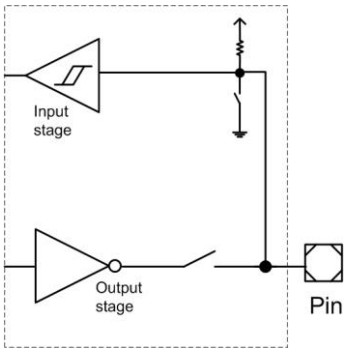
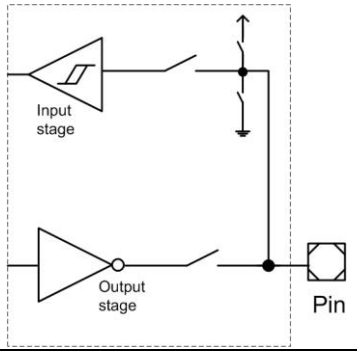
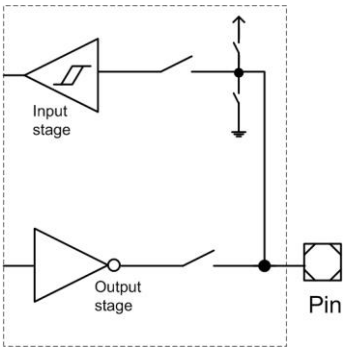
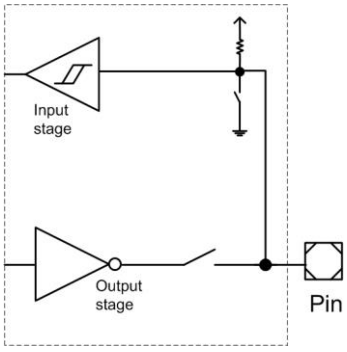
Table 6 Port/Pin Configuration for Debug mode (SPD)

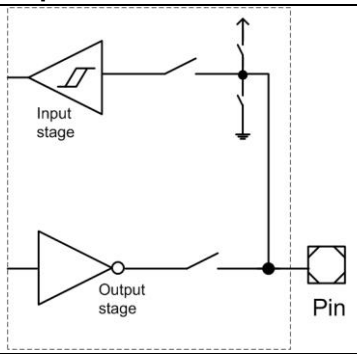
Port/Pin	During Power up	After Reset (Low)	During Debug mode (SPD)
P0.14 SPD_0 for Debug mode (SPD) P1.3 SPD_1 for Debug mode (SPD)			

7.5 ASC BSL half-duplex mode

Table 7 shows the Port pins used for the ASC BSL half-duplex mode and the corresponding pin configuration. The other Port pins not referenced in the table follow the configuration given for User Productive mode (see **Table 3**).

Table 7 Port/Pin Configuration for ASC BSL half-duplex mode

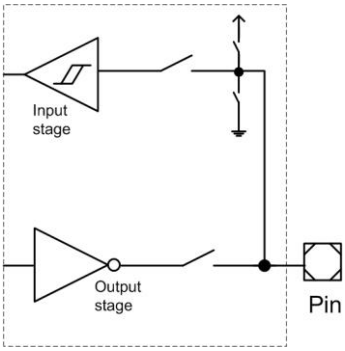
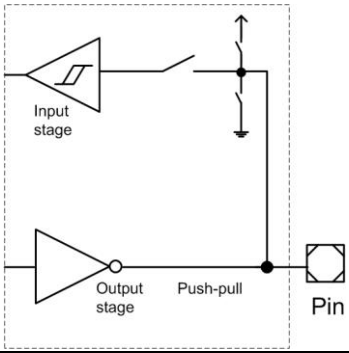
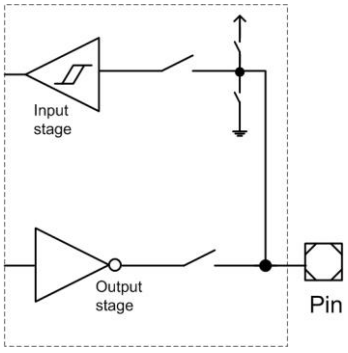
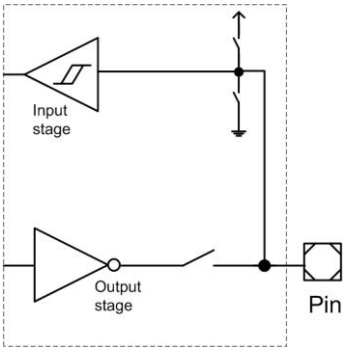
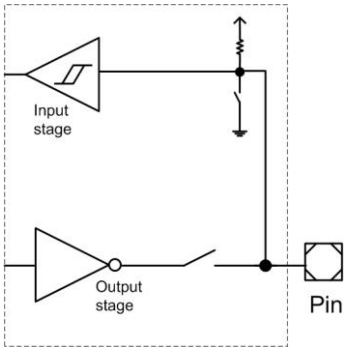
Port/Pin	During Power up	After Reset (Low)	During ASC BSL half-duplex mode
P0.14 RX/TX for ASC BSL half-duplex mode			If channel 0 is selected,  Otherwise, 
P1.3 RX/TX for ASC BSL half-duplex mode			If channel 1 is selected,  Otherwise,

Port/Pin	During Power up	After Reset (Low)	During ASC BSL half-duplex mode
			

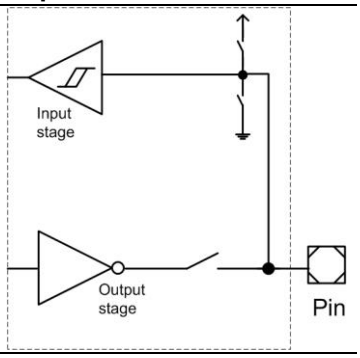
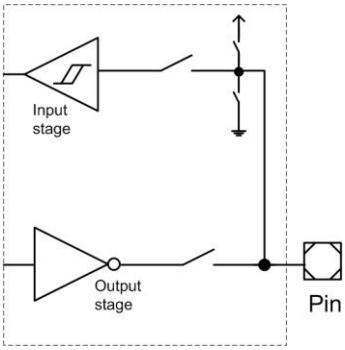
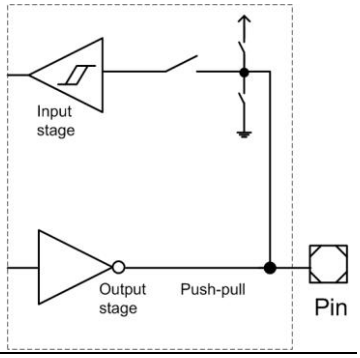
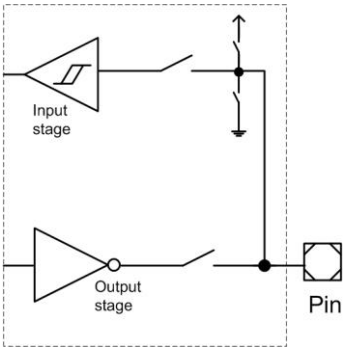
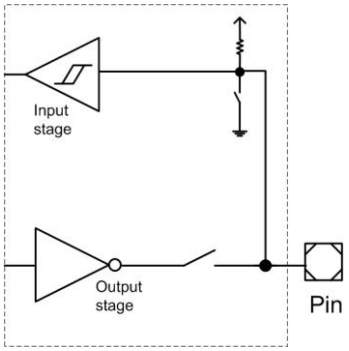
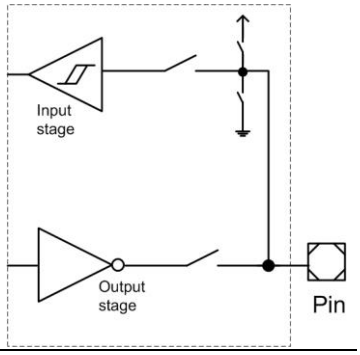
7.6 ASC BSL full-duplex mode

Table 8 shows the Port pins used for the ASC BSL full-duplex mode and the corresponding pin configuration. The other Port pins not referenced in the table follow the configuration given for User Productive mode (see **Table 3**).

Table 8 Port/Pin Configuration for ASC BSL full-duplex mode

Port/Pin	During Power up	After Reset (Low)	During ASC BSL full-duplex mode
P0.14 RX for ASC BSL full-duplex mode			If channel 0 is selected, 
P0.15 TX for ASC BSL full-duplex mode			If channel 0 is selected,  Otherwise,

Port **Configuration during Power-up until end of Startup Software (SSW) Execution**

Port/Pin	During Power up	After Reset (Low)	During ASC BSL full-duplex mode
			
P1.2 TX for ASC BSL full-duplex mode			If channel 1 is selected, 
P1.3 RX for ASC BSL full-duplex mode			If channel 1 is selected,  Otherwise, 

w w w . i n f i n e o n . c o m

Published by Infineon Technologies AG