ATmega128

Lock Bits & Fuse Bits

System Clock

I/O Ports

JTAG

Lock Bits

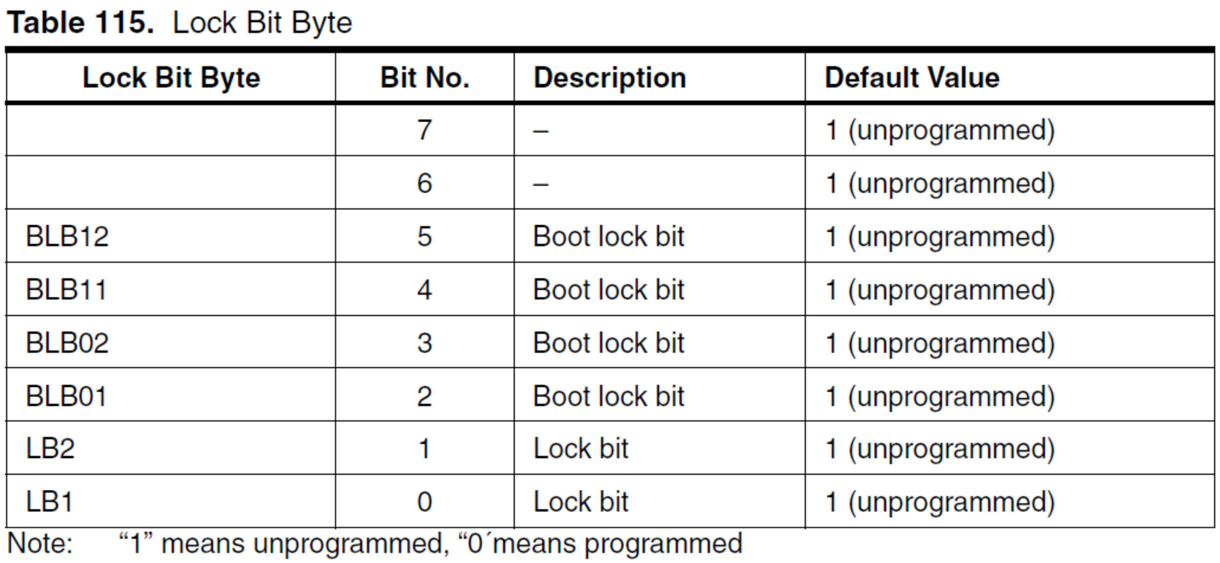


Table 115. Lock Bit Byte

Lock Bit Byte	Bit No.	Description	Default Value
	7	–	1 (unprogrammed)
	6	–	1 (unprogrammed)
BLB12	5	Boot lock bit	1 (unprogrammed)
BLB11	4	Boot lock bit	1 (unprogrammed)
BLB02	3	Boot lock bit	1 (unprogrammed)
BLB01	2	Boot lock bit	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

Note: “1” means unprogrammed, “0´means programmed

Protect Boot Loader Section

Protect Application Section

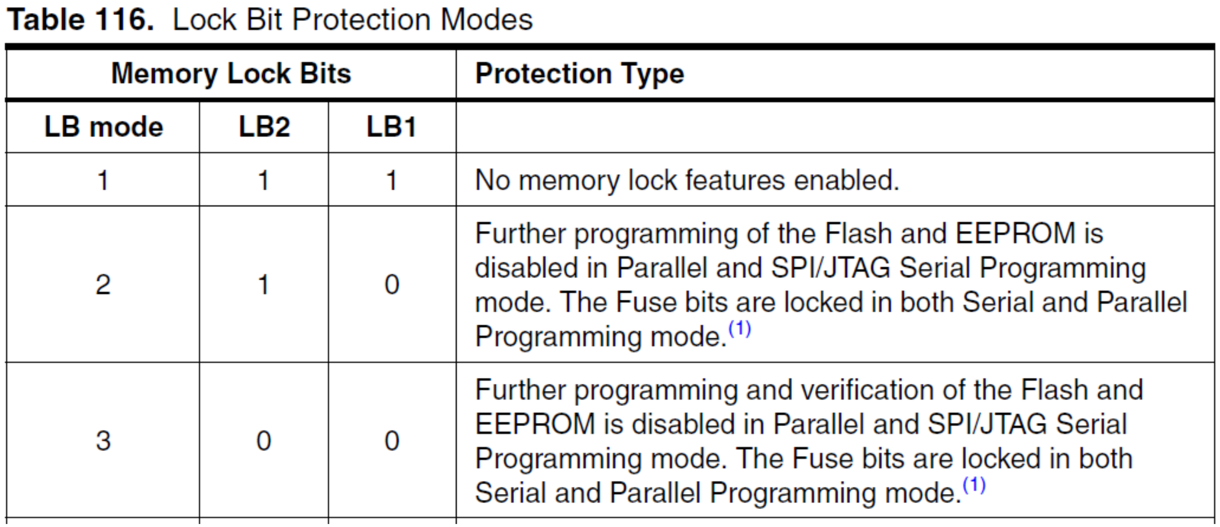


Table 116. Lock Bit Protection Modes

Memory Lock Bits			Protection Type
LB mode	LB2	LB1	
1	1	1	No memory lock features enabled.
2	1	0	Further programming of the Flash and EEPROM is disabled in Parallel and SPI/JTAG Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.[1]
3	0	0	Further programming and verification of the Flash and EEPROM is disabled in Parallel and SPI/JTAG Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.[1]

Write Protection

Read/Write Protection

Fuse Bits: Extended Fuse Byte

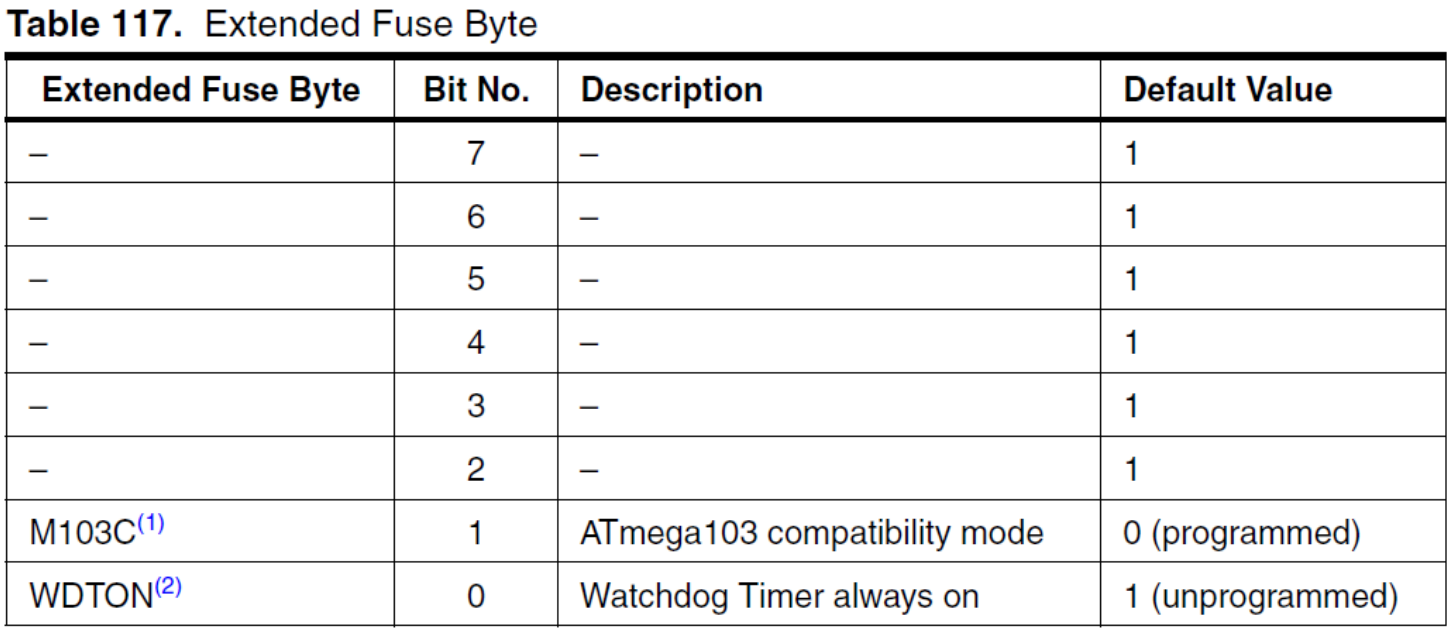
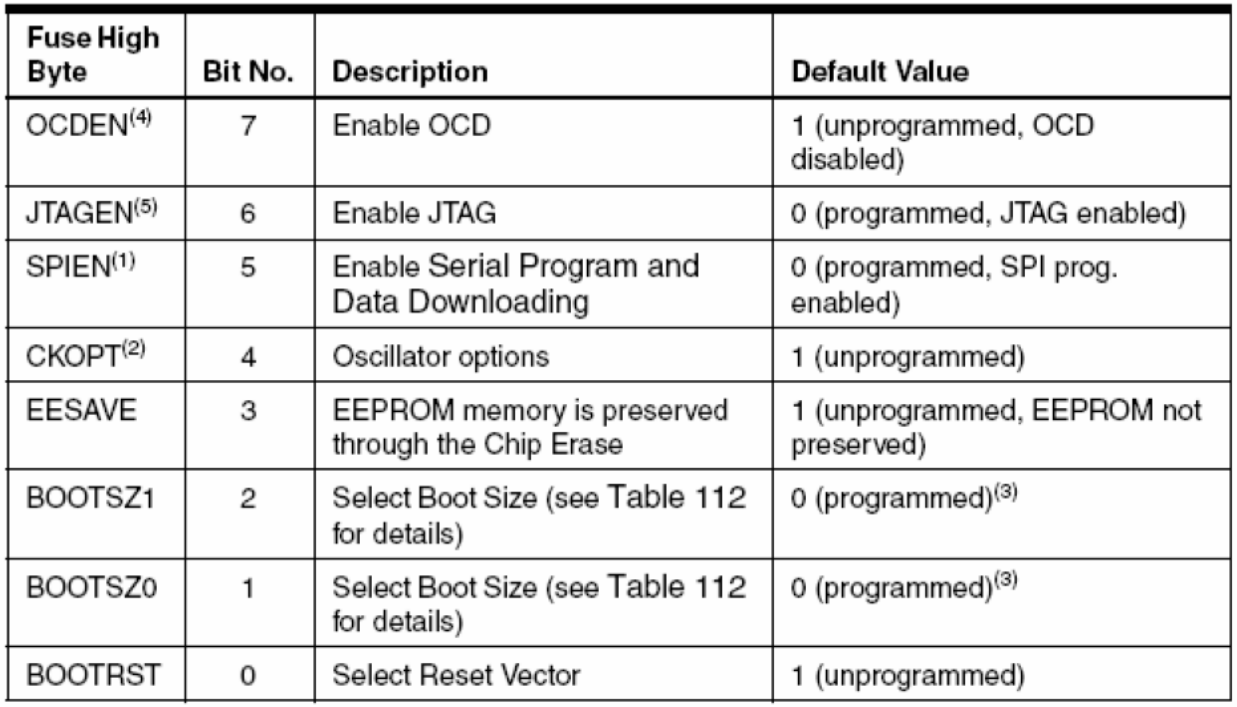


Table 117. Extended Fuse Byte

Extended Fuse Byte	Bit No.	Description	Default Value
–	7	–	1
–	6	–	1
–	5	–	1
–	4	–	1
–	3	–	1
–	2	–	1
M103C[(1)]	1	ATmega103 compatibility mode	0 (programmed)
WDTON[(2)]	0	Watchdog Timer always on	1 (unprogrammed)

Fuse Bits: Fuse High Bytes

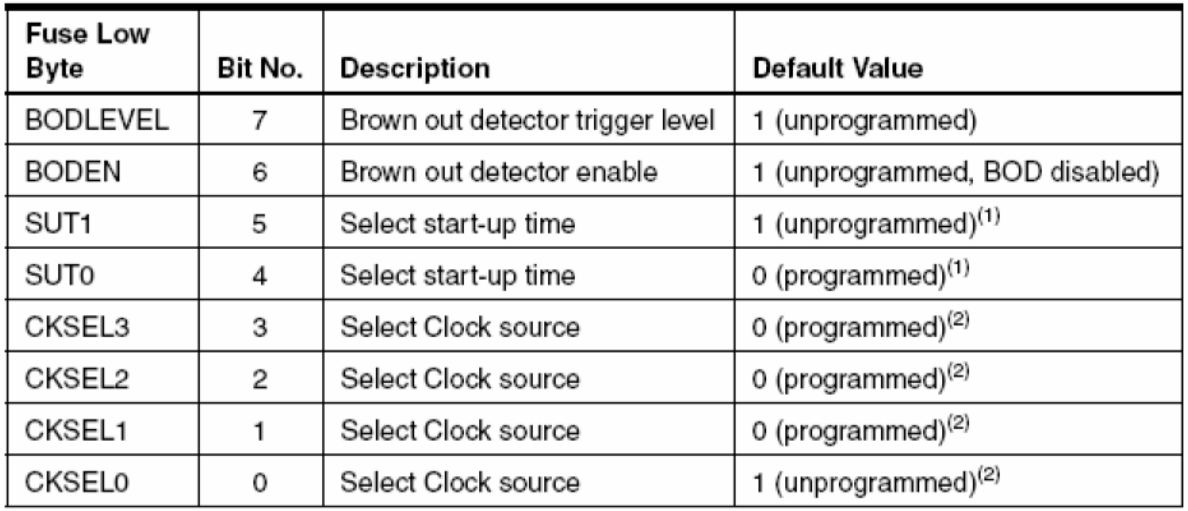


Fuse High Byte	Bit No.	Description	Default Value
OCDEN[4]	7	Enable OCD	1 (unprogrammed, OCD disabled)
JTAGEN[5]	6	Enable JTAG	0 (programmed, JTAG enabled)
SPIEN[1]	5	Enable Serial Program and Data Downloading	0 (programmed, SPI prog. enabled)
CKOPT[2]	4	Oscillator options	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed, EEPROM not preserved)
BOOTSZ1	2	Select Boot Size (see Table 112 for details)	0 (programmed)[3]
BOOTSZ0	1	Select Boot Size (see Table 112 for details)	0 (programmed)[3]
BOOTRST	0	Select Reset Vector	1 (unprogrammed)

OCDR Register Access

Amplitude Adjust.

Fuse Bits: Fuse Low Byte



Fuse Low Byte	Bit No.	Description	Default Value
BODLEVEL	7	Brown out detector trigger level	1 (unprogrammed)
BODEN	6	Brown out detector enable	1 (unprogrammed, BOD disabled)
SUT1	5	Select start-up time	1 (unprogrammed)[1]
SUT0	4	Select start-up time	0 (programmed)[1]
CKSEL3	3	Select Clock source	0 (programmed)[2]
CKSEL2	2	Select Clock source	0 (programmed)[2]
CKSEL1	1	Select Clock source	0 (programmed)[2]
CKSEL0	0	Select Clock source	1 (unprogrammed)[2]

Power Level Detector

1: 2.7V 0:4.0V

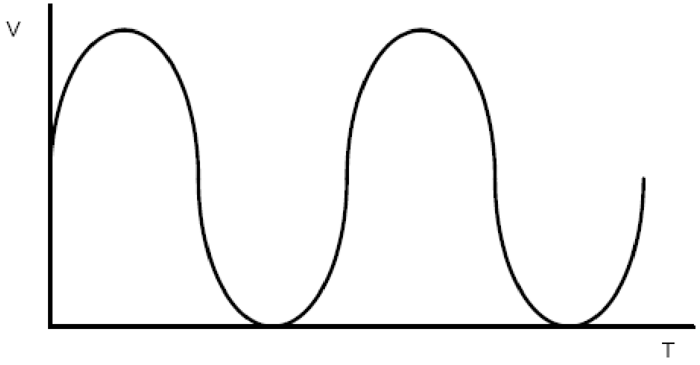
Start Up time

Clock Source Selection

Fuse Bits

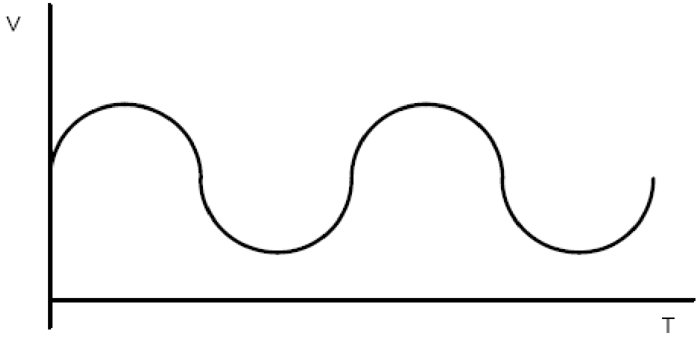
Clock Point 1 - CKOPT

The CKOPT fuse selects between two different Oscillator Amplifier modes



CKOPT = 0

Max = 16Mhz

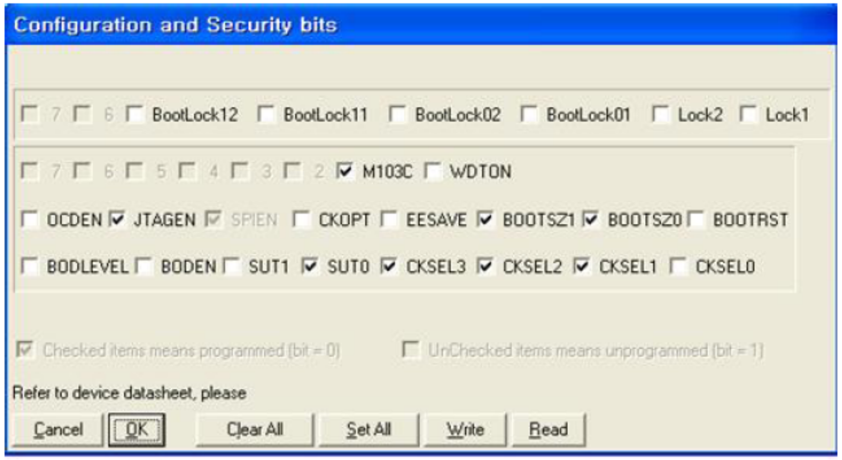


CKOPT = 1

Max = 8Mhz

PonyProg2000(obsolete)

PonyProg 2000 Setting



Configuration and Security bits

□ 7 □ 6 □ BootLock12 □ BootLock11 □ BootLock02 □ BootLock01 □ Lock2 □ Lock1

□ 7 □ 6 □ 5 □ 4 □ 3 □ 2 ☑ M103C □ WDTON

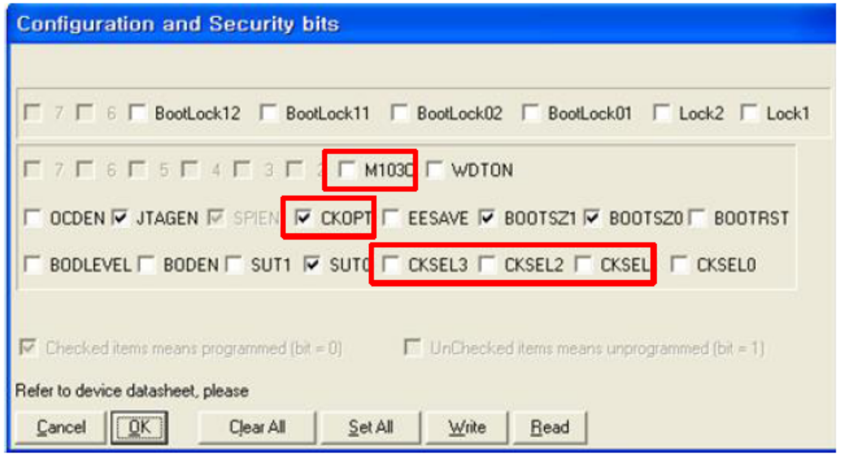
□ OCDEN ☑ JTAGEN ☑ SPIEN □ CKOPT □ EESAVE ☑ BOOTSZ1 ☑ BOOTSZ0 □ BOOTRST

□ BODLEVEL □ BODEN □ SUT1 ☑ SUT0 ☑ CKSEL3 ☑ CKSEL2 ☑ CKSEL1 □ CKSEL0

☑ Checked items means programmed (bit = 0) □ UnChecked items means unprogrammed (bit = 1)

Refer to device datasheet, please

Default Status



Configuration and Security bits

□ 7 □ 6 □ BootLock12 □ BootLock11 □ BootLock02 □ BootLock01 □ Lock2 □ Lock1

□ 7 □ 6 □ 5 □ 4 □ 3 □ □ M103C □ WDTON

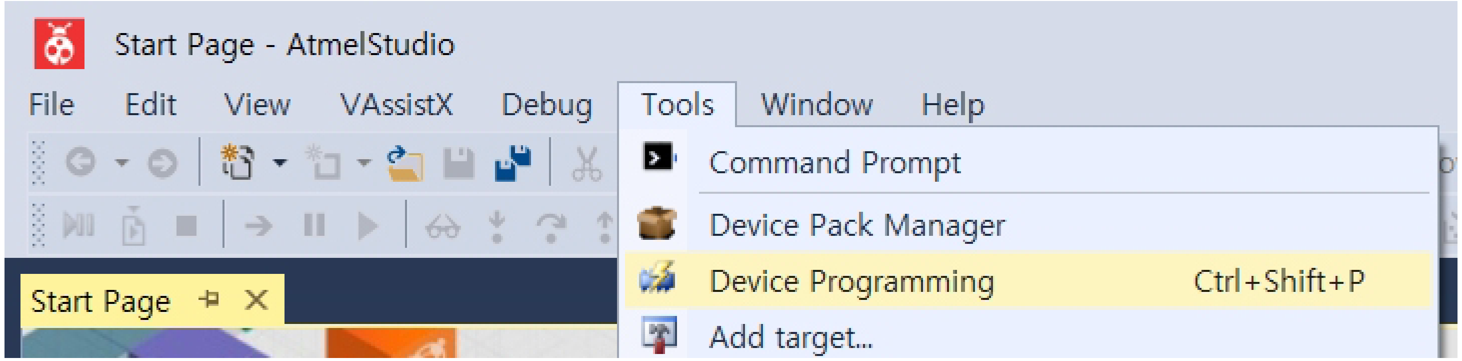
□ OCDEN ☑ JTAGEN ☑ SPIEN ☑ CKOPT □ EESAVE ☑ BOOTSZ1 ☑ BOOTSZ0 □ BOOTRST

□ BODLEVEL □ BODEN □ SUT1 ☑ SUT0 □ CKSEL3 □ CKSEL2 □ CKSEL □ CKSEL0

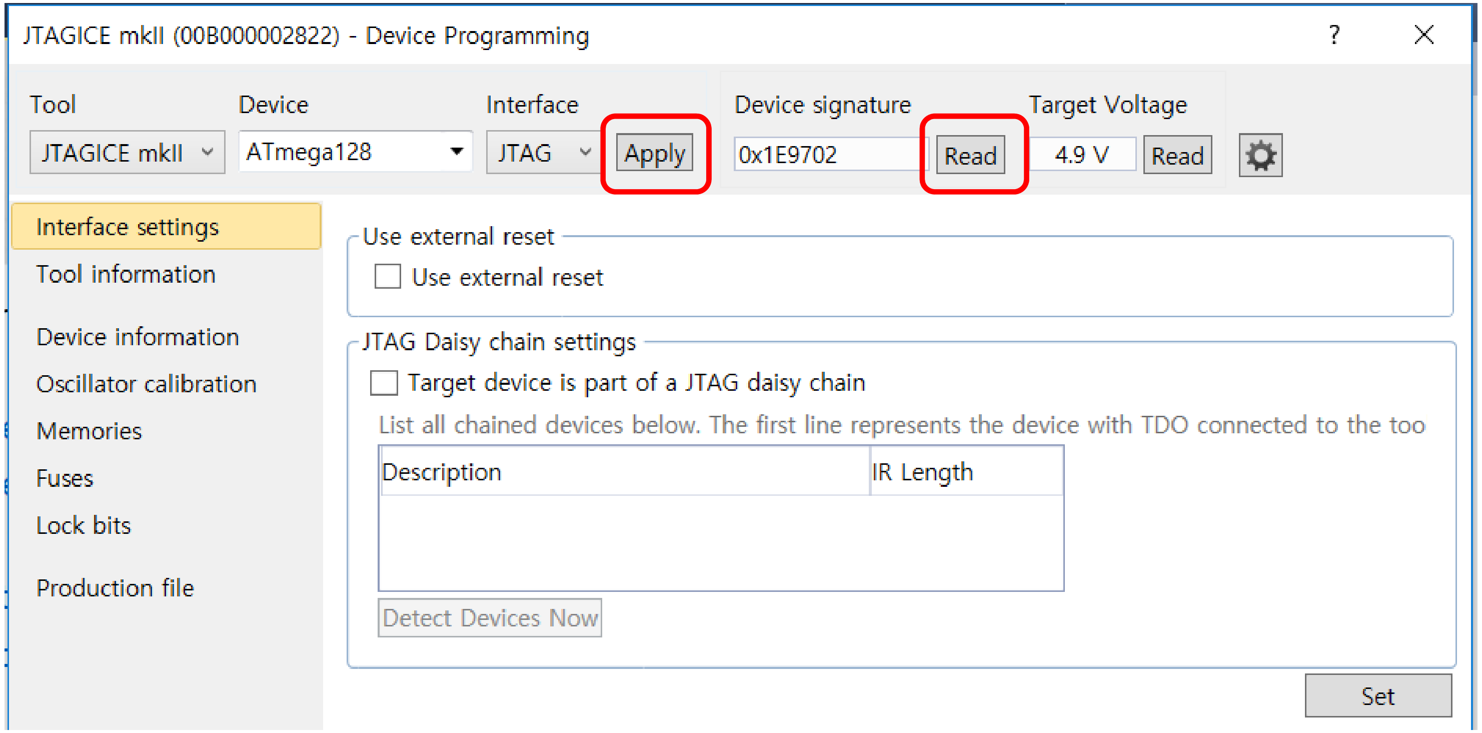
☑ Checked items means programmed (bit = 0) □ UnChecked items means unprogrammed (bit = 1)

Refer to device datasheet, please

Atmel Studio



Start Page - AtmelStudio



JTAGICE mkII (00B000002822) - Device Programming

Tool: JTAGICE mkII
Device: ATmega128
Interface: JTAG
Device signature: 0x1E9702
Target Voltage: 4.9 V

Use external reset

☐ Use external reset

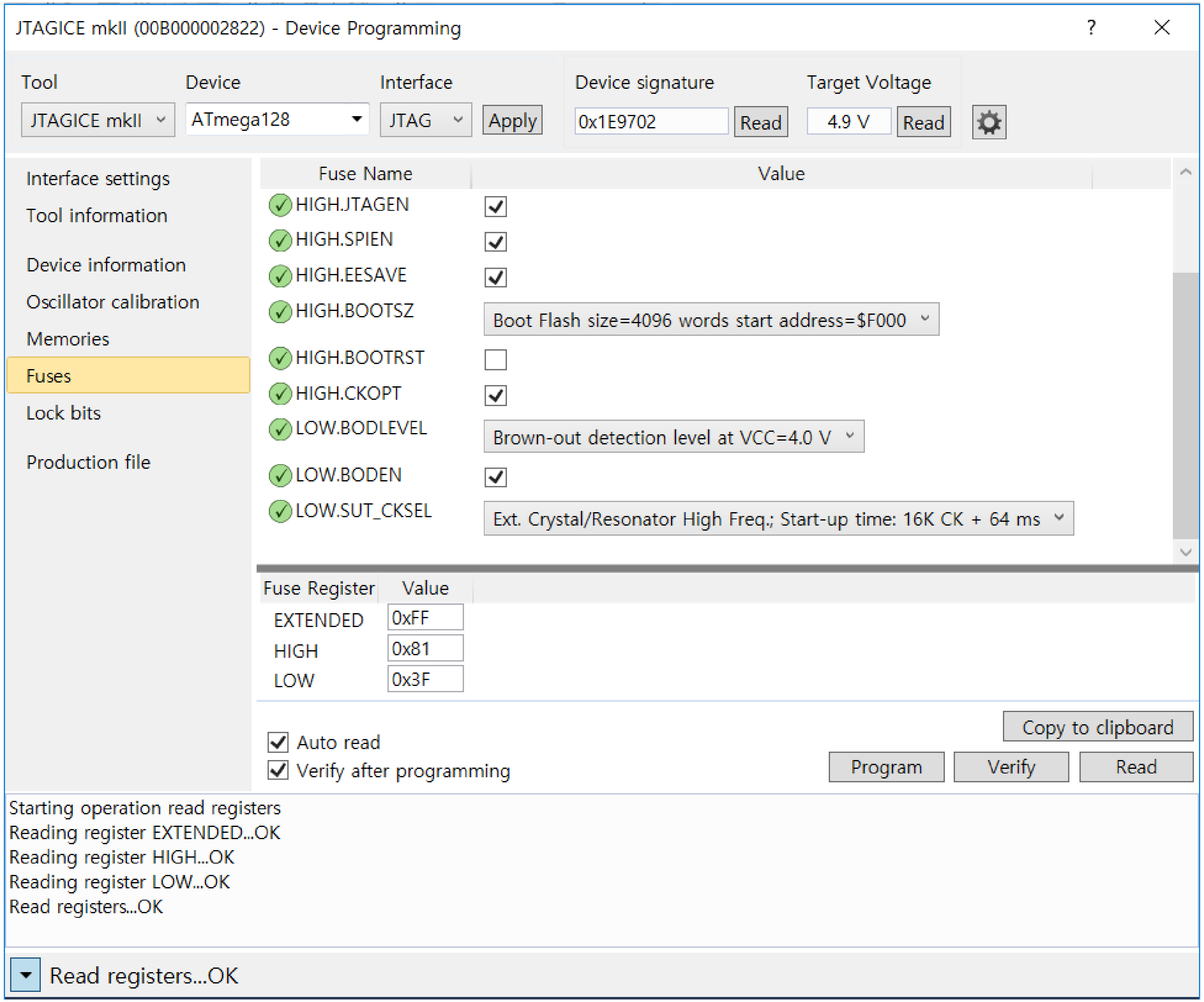
JTAG Daisy chain settings

☐ Target device is part of a JTAG daisy chain

List all chained devices below. The first line represents the device with TDO connected to the too

Description	IR Length

Fuses



JTAGICE mkII (00B000002822) - Device Programming

Tool: JTAGICE mkII
Device: ATmega128
Interface: JTAG
Device signature: 0x1E9702
Target Voltage: 4.9 V

Fuse Name	Value
HIGH.JTAGEN	☑
HIGH.SPIEN	☑
HIGH.EESAVE	☑
HIGH.BOOTSZ	Boot Flash size=4096 words start address=$F000
HIGH.BOOTRST	□
HIGH.CKOPT	☑
LOW.BODLEVEL	Brown-out detection level at VCC=4.0 V
LOW.BODEN	☑
LOW.SUT_CKSEL	Ext. Crystal/Resonator High Freq.; Start-up time: 16K CK + 64 ms

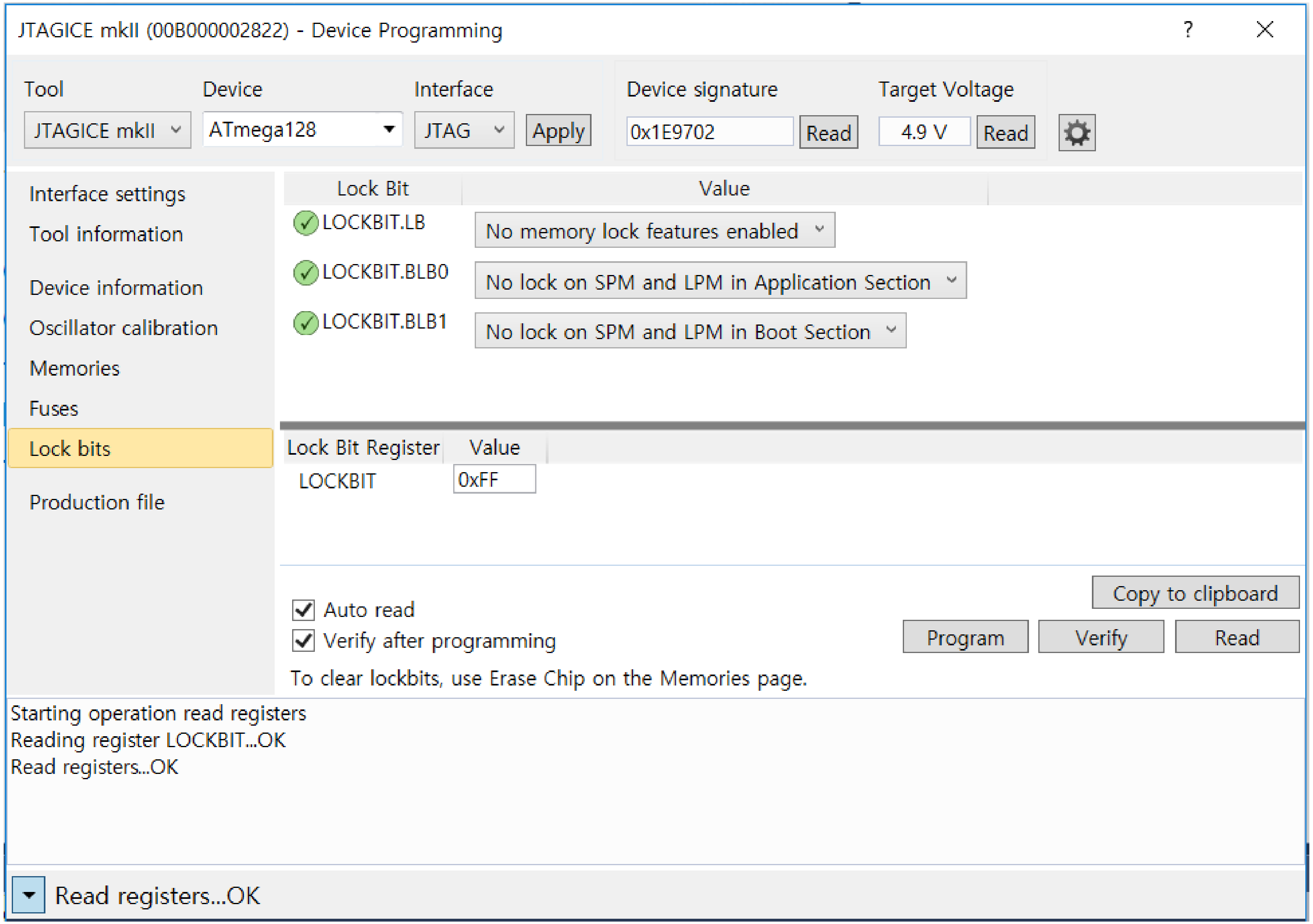
Fuse Register	Value
EXTENDED	0xFF
HIGH	0x81
LOW	0x3F

☑ Auto read
☑ Verify after programming

```
Starting operation read registers
Reading register EXTENDED...OK
Reading register HIGH...OK
Reading register LOW...OK
Read registers...OK
```

Read registers...OK

Lock bits



JTAGICE mkII (00B000002822) - Device Programming

Tool: JTAGICE mkII
Device: ATmega128
Interface: JTAG
Device signature: 0x1E9702
Target Voltage: 4.9 V

Lock Bit	Value
✓ LOCKBIT.LB	No memory lock features enabled
✓ LOCKBIT.BLB0	No lock on SPM and LPM in Application Section
✓ LOCKBIT.BLB1	No lock on SPM and LPM in Boot Section

Lock Bit Register	Value
LOCKBIT	0xFF

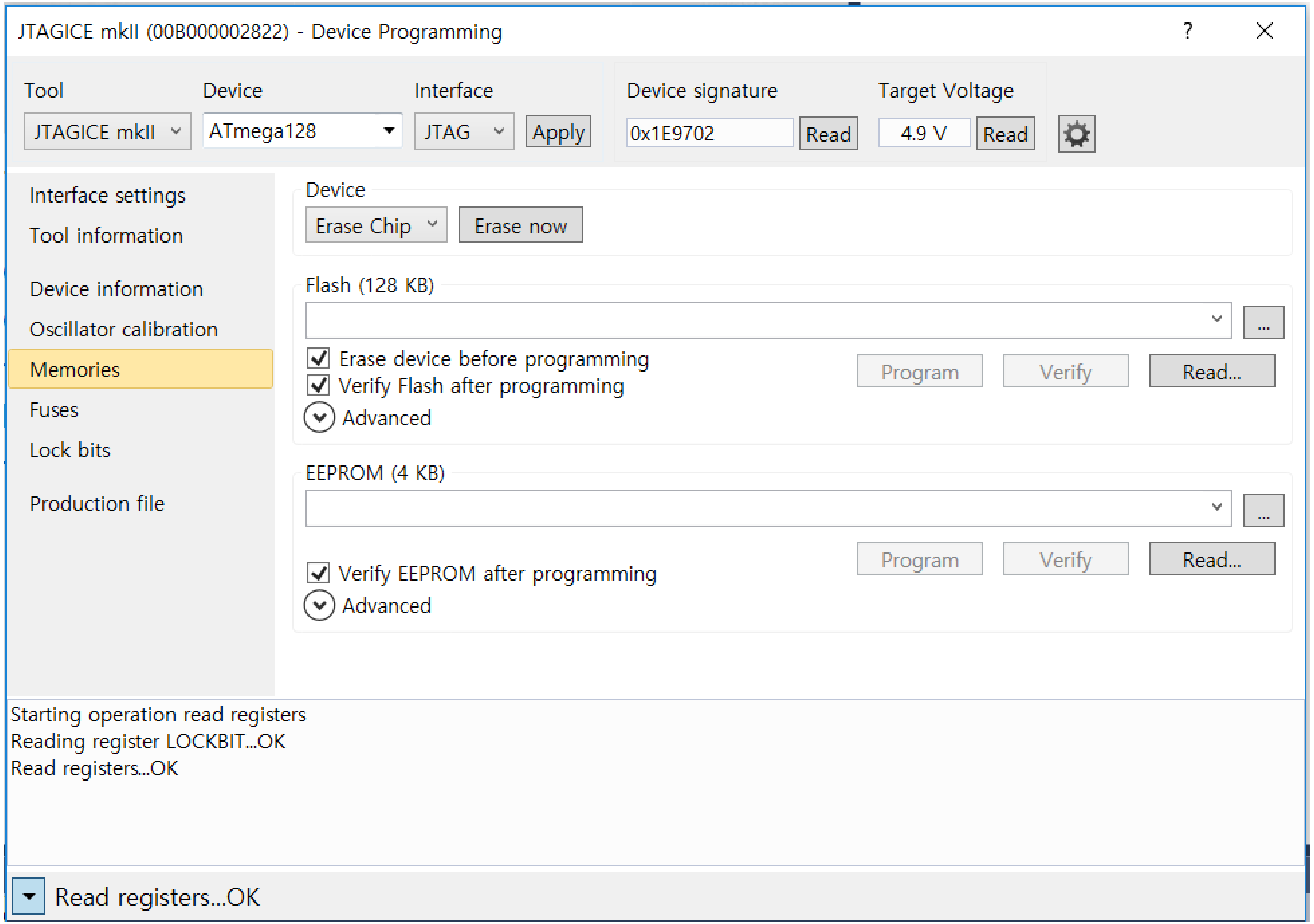
☑ Auto read
☑ Verify after programming

To clear lockbits, use Erase Chip on the Memories page.

```
Starting operation read registers
Reading register LOCKBIT...OK
Read registers...OK
```

Read registers...OK

Memories



JTAGICE mkII (00B000002822) - Device Programming

Tool: JTAGICE mkII

Device: ATmega128

Interface: JTAG

Device signature: 0x1E9702

Target Voltage: 4.9 V

☑ Erase device before programming

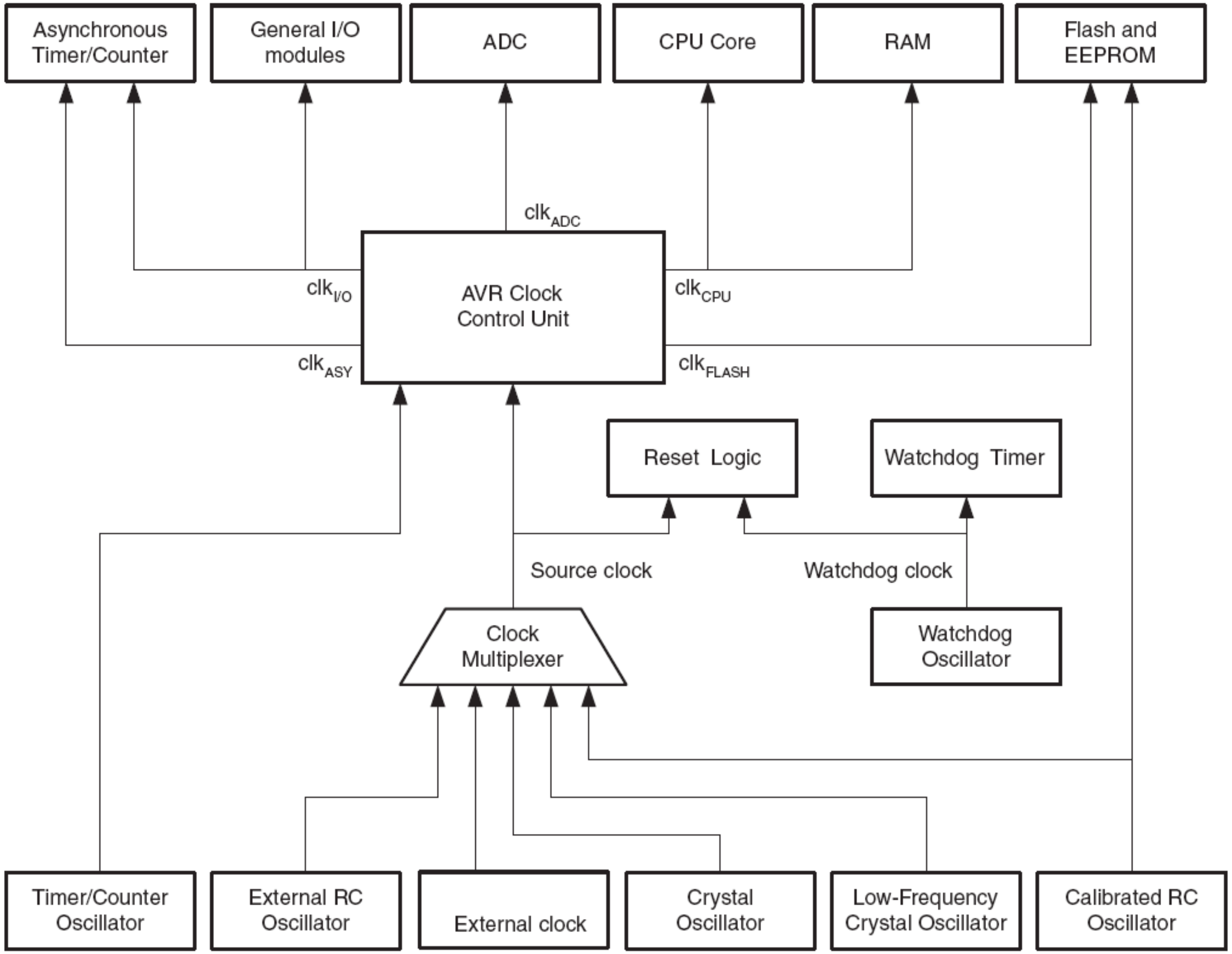
☑ Verify Flash after programming

☑ Verify EEPROM after programming

```
Starting operation read registers
Reading register LOCKBIT...OK
Read registers...OK
```

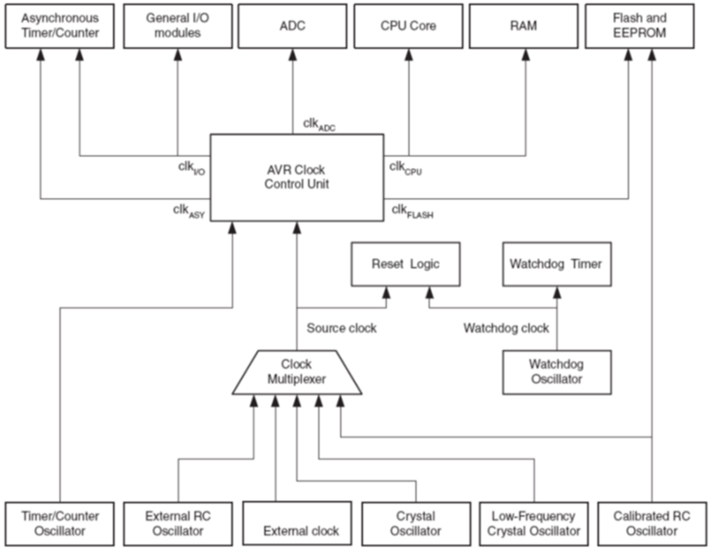
Read registers...OK

System Clock



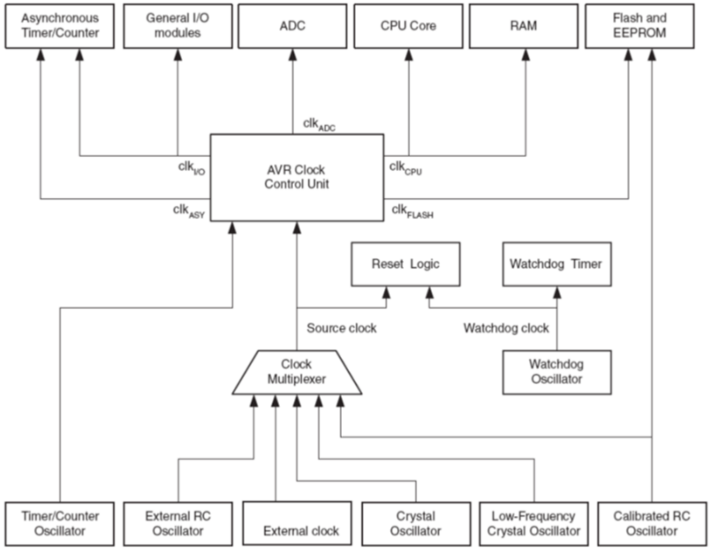
System Clock Overview

- **CPU Clock – $\mathrm{clk}_{\mathrm{CPU}}$**
  - The CPU clock is routed to parts of the system concerned with operation of the AVR core.
  - Examples of such modules are the General Purpose Register File, the Status Register and the data memory holding the Stack Pointer.
  - Halting the CPU clock inhibits the core from performing general operations and calculations.



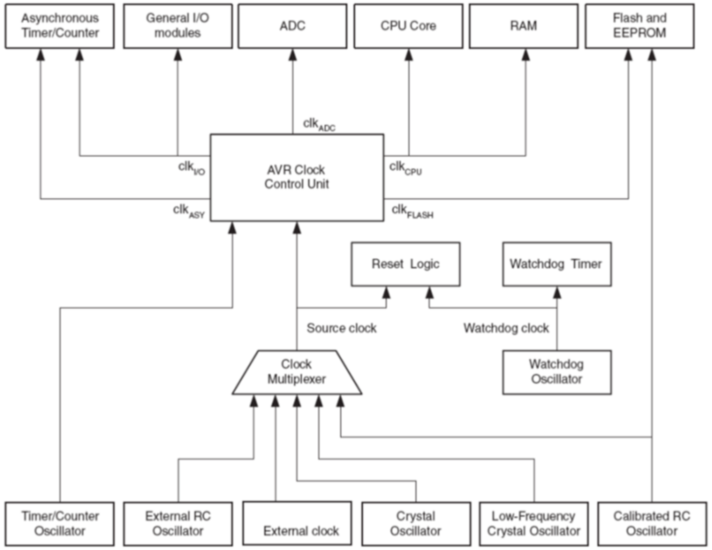
System Clock Overview

- **I/O Clock – clk$_{I/O}$**
  - The I/O clock is used by the majority of the I/O modules, like Timer/Counters, SPI, and USART.
  - The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted. Also note that address recognition in the TWI module is carried out asynchronously when clkI/O is halted, enabling TWI address reception in all sleep modes.



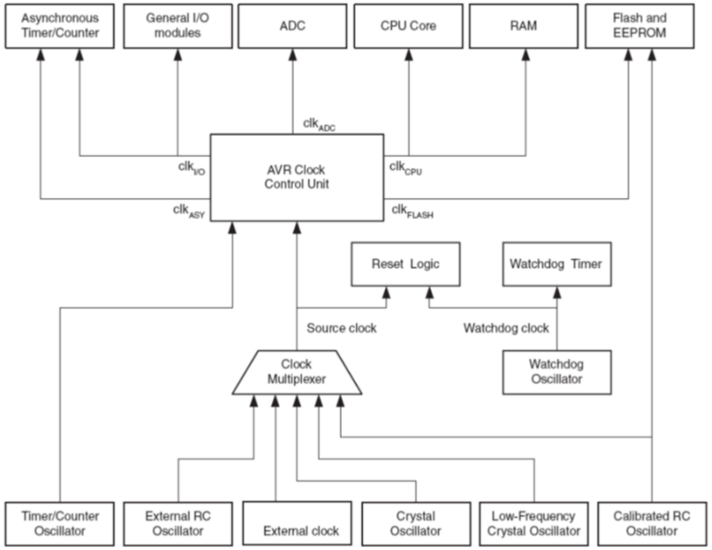
System Clock Overview

- **Flash Clock – clk$_{FLASH}$**
  - The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.
- **Async Timer Clock – clk$_{ASY}$**
  - Asynchronous Timer clock allows the Asynchronous Timer/Counter to be clocked directly from an external 32 kHz clock crystal.
  - The dedicated clock domain allows using this Timer/Counter as a real-time counter even when the device is in sleep mode.



System Clock Overview

- **ADC Clock – clk_{ADC}**
  - The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry.
  - This gives more accurate ADC conversion results.



Clock Sources

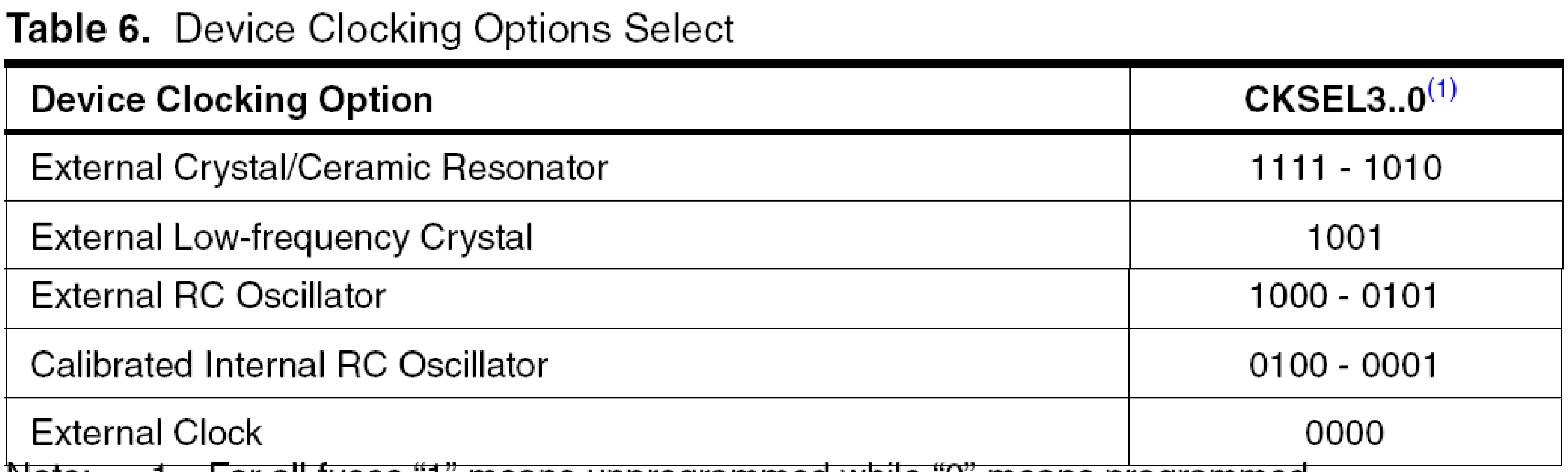


Table 6. Device Clocking Options Select

Device Clocking Option	CKSEL3..0[1]
External Crystal/Ceramic Resonator	1111 - 1010
External Low-frequency Crystal	1001
External RC Oscillator	1000 - 0101
Calibrated Internal RC Oscillator	0100 - 0001
External Clock	0000

Note: 1. For all fuses “1” means unprogrammed while “0” means programmed.

Default Clock Source

The device is shipped with CKSEL = “0001” and SUT = “10”. The default clock source setting is therefore the Internal RC Oscillator with longest startup time. This default setting ensures that all users can make their desired clock source setting using an In-System or Parallel Programmer.

Crystal Oscillator

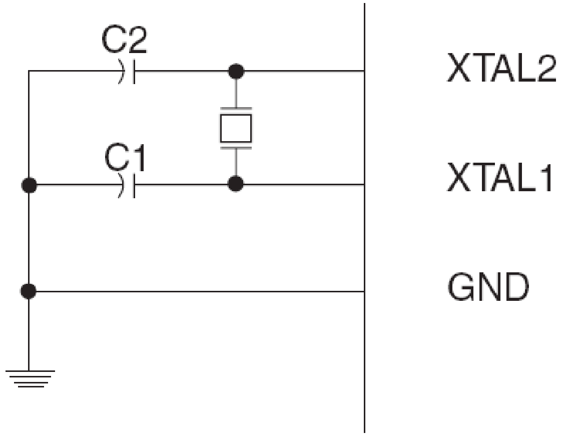


Table 8. Crystal Oscillator Operating Modes

CKOPT	CKSEL3..1	Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals
1	101[1]	0.4 - 0.9	–
1	110	0.9 - 3.0	12 pF - 22 pF
1	111	3.0 - 8.0	12 pF - 22 pF
0	101, 110, 111	1.0 -	12 pF - 22 pF

Note: 1. This option should not be used with crystals, only with ceramic resonators.

Crystal Oscillator

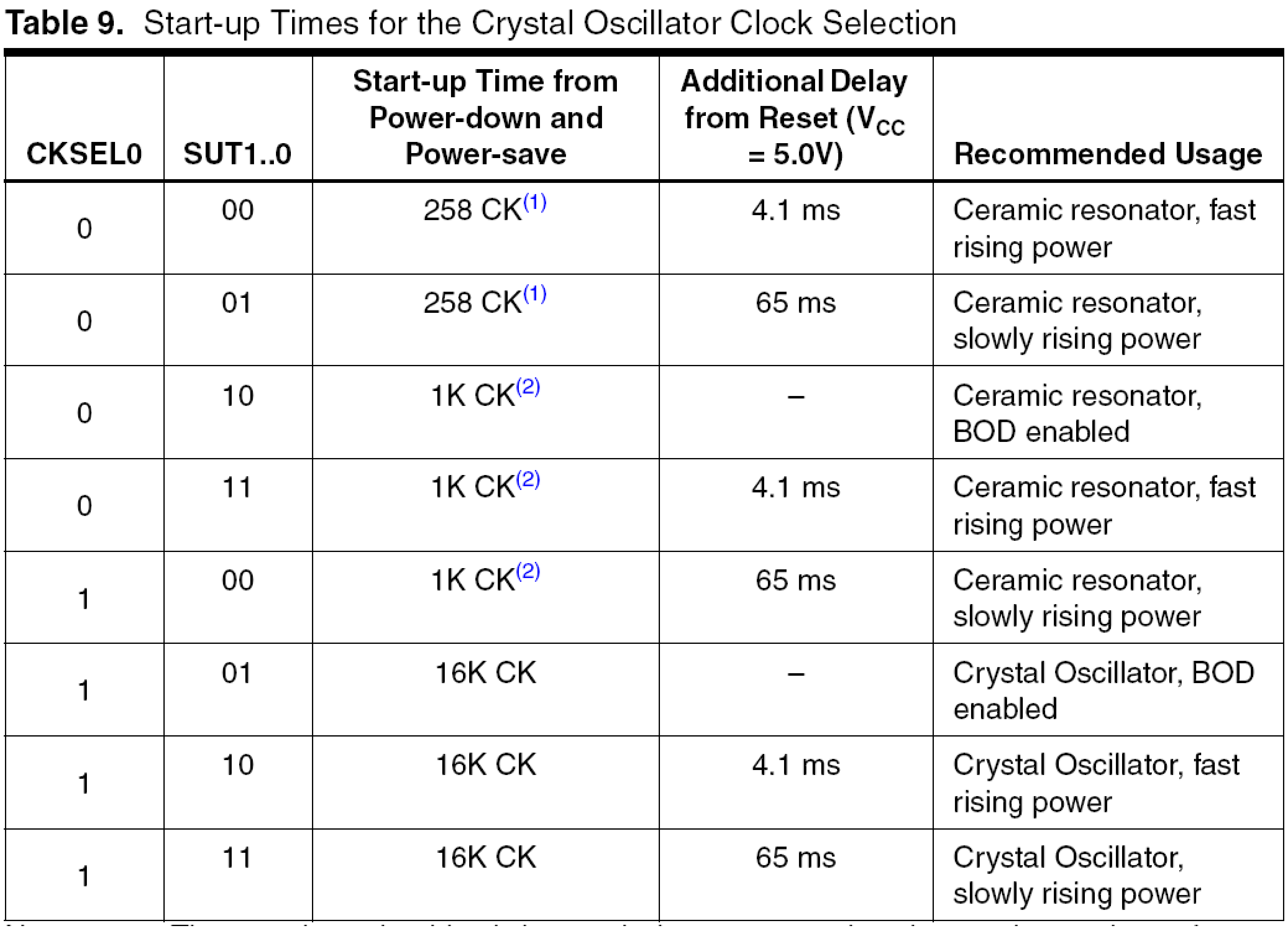
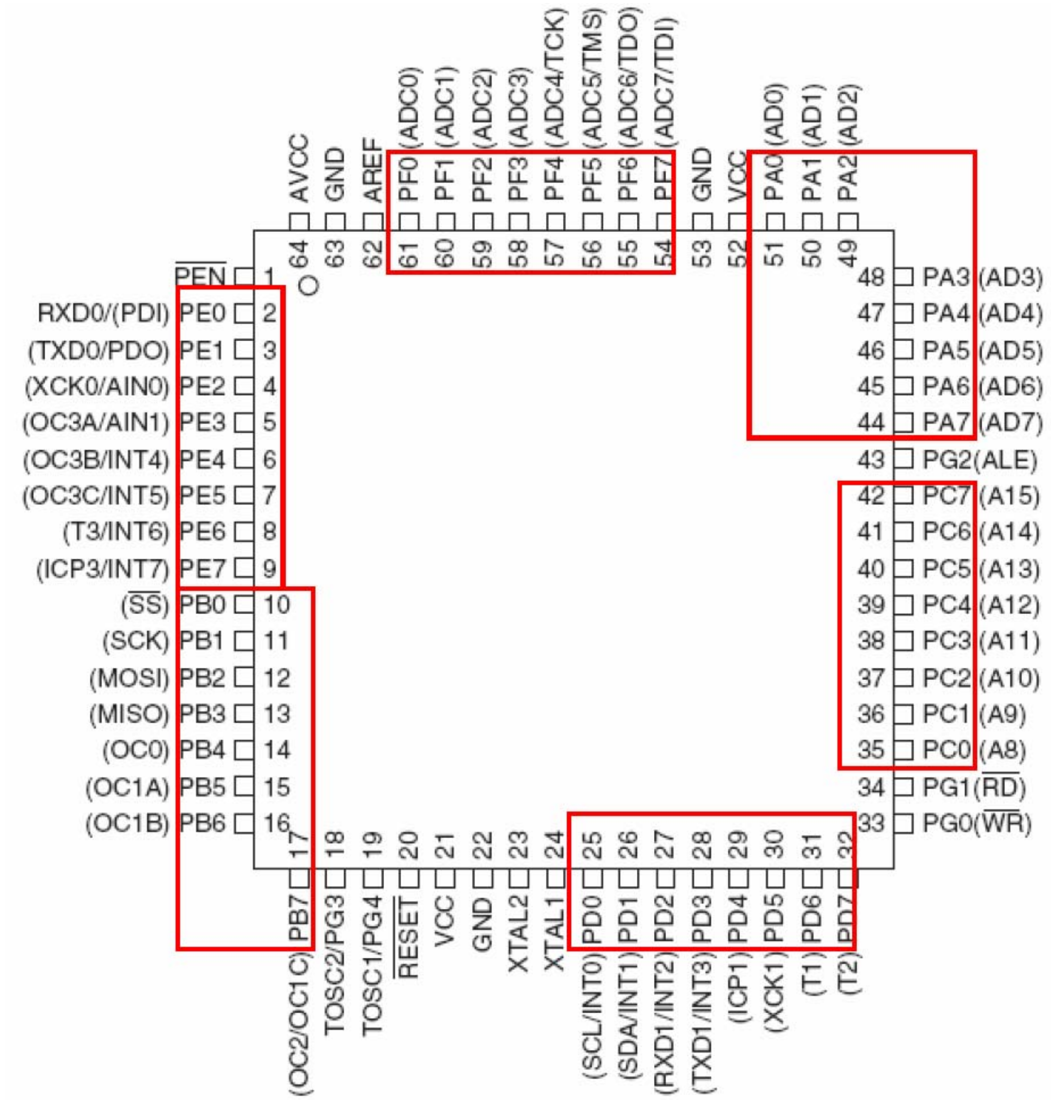


Table 9. Start-up Times for the Crystal Oscillator Clock Selection

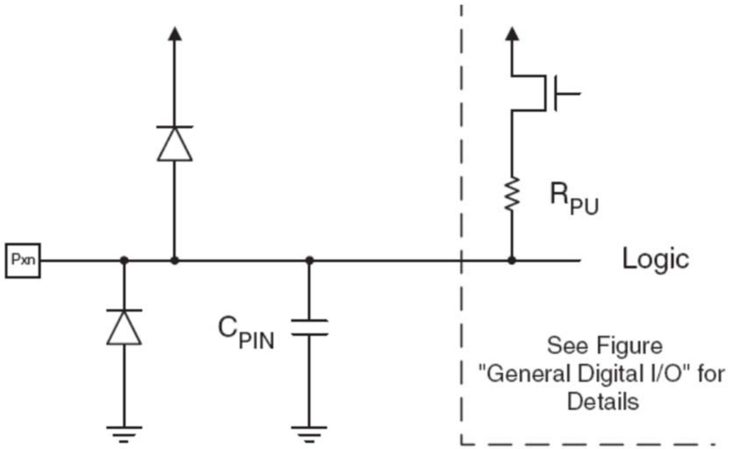
CKSEL0	SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
0	00	258 CK[1]	4.1 ms	Ceramic resonator, fast rising power
0	01	258 CK[1]	65 ms	Ceramic resonator, slowly rising power
0	10	1K CK[2]	–	Ceramic resonator, BOD enabled
0	11	1K CK[2]	4.1 ms	Ceramic resonator, fast rising power
1	00	1K CK[2]	65 ms	Ceramic resonator, slowly rising power
1	01	16K CK	–	Crystal Oscillator, BOD enabled
1	10	16K CK	4.1 ms	Crystal Oscillator, fast rising power
1	11	16K CK	65 ms	Crystal Oscillator, slowly rising power

I/O Ports


PEN 1
RXD0/(PDI) PE0 2
(TXD0/PDO) PE1 3
(XCK0/AIN0) PE2 4
(OC3A/AIN1) PE3 5
(OC3B/INT4) PE4 6
(OC3C/INT5) PE5 7
(T3/INT6) PE6 8
(ICP3/INT7) PE7 9
(SS) PB0 10
(SCK) PB1 11
(MOSI) PB2 12
(MISO) PB3 13
(OC0) PB4 14
(OC1A) PB5 15
(OC1B) PB6 16
(OC2/OC1C) PB7 17
TOSC2/PG3 18
TOSC1/PG4 19
RESET 20
VCC 21
GND 22
XTAL2 23
XTAL1 24
(SCL/INT0) PD0 25
(SDA/INT1) PD1 26
(RXD1/INT2) PD2 27
(TXD1/INT3) PD3 28
(ICP1) PD4 29
(XCK1) PD5 30
(T1) PD6 31
(T2) PD7 32
33 PG0(WR)
34 PG1(RD)
35 PC0 (A8)
36 PC1 (A9)
37 PC2 (A10)
38 PC3 (A11)
39 PC4 (A12)
40 PC5 (A13)
41 PC6 (A14)
42 PC7 (A15)
43 PG2(ALE)
44 PA7 (AD7)
45 PA6 (AD6)
46 PA5 (AD5)
47 PA4 (AD4)
48 PA3 (AD3)
49 PA2 (AD2)
50 PA1 (AD1)
51 PA0 (AD0)
52 VCC
53 GND
54 PF7 (ADC7/TDI)
55 PF6 (ADC6/TDO)
56 PF5 (ADC5/TMS)
57 PF4 (ADC4/TCK)
58 PF3 (ADC3)
59 PF2 (ADC2)
60 PF1 (ADC1)
61 PF0 (ADC0)
62 AREF
63 GND
64 AVCC

I/O Ports

- **Three I/O memory address locations allocated for each port**
  - Data Register – PORTx
  - Data Direction Register – DDRx
  - Port Input Pins – PINx

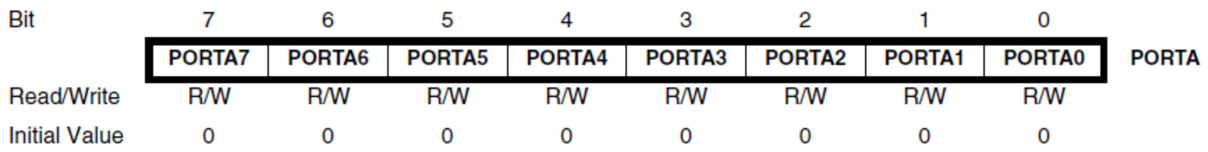


- PINx : read only
- PORTx, DDRx : read/write
- Programmable Pull-up Register
  - Pull-up Disable: PUD bit in SFIOR disables the pull-up function for all pins in all ports

I/O Port Registers

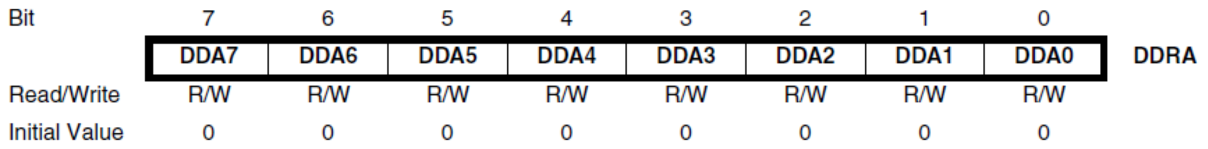
- **Example for Port A**

Port A Data Register – PORTA



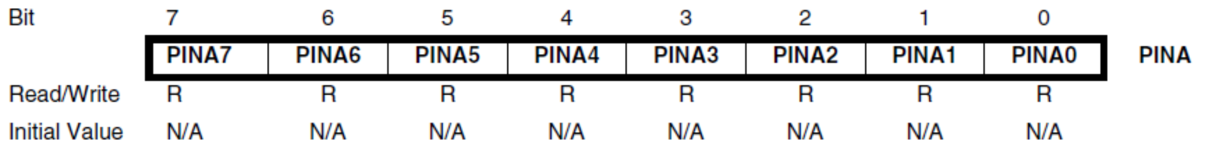
Bit	7	6	5	4	3	2	1	0	
	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Data Direction Register – DDRA



Bit	7	6	5	4	3	2	1	0	
	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Input Pins Address – PINA

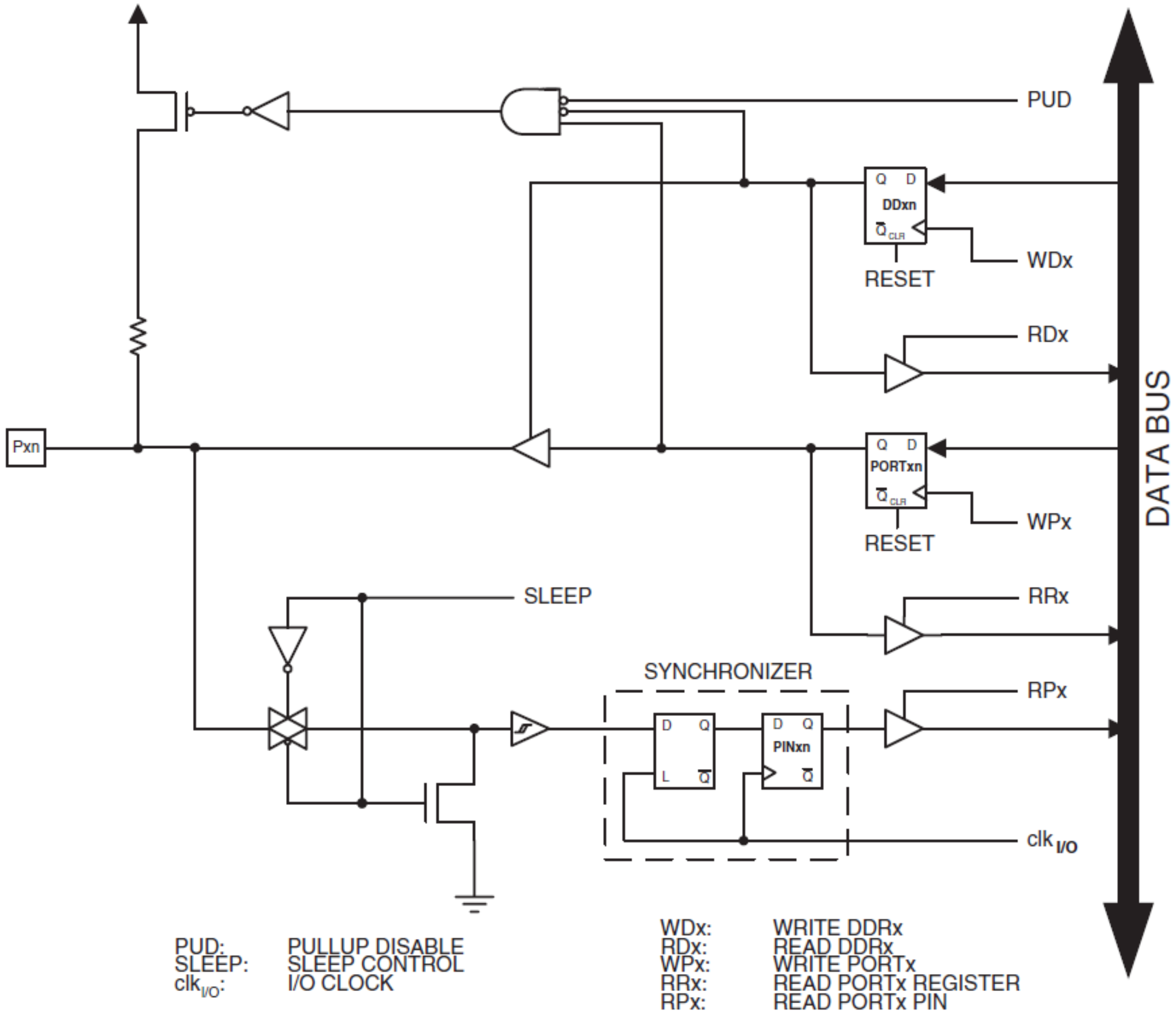


Bit	7	6	5	4	3	2	1	0	
	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

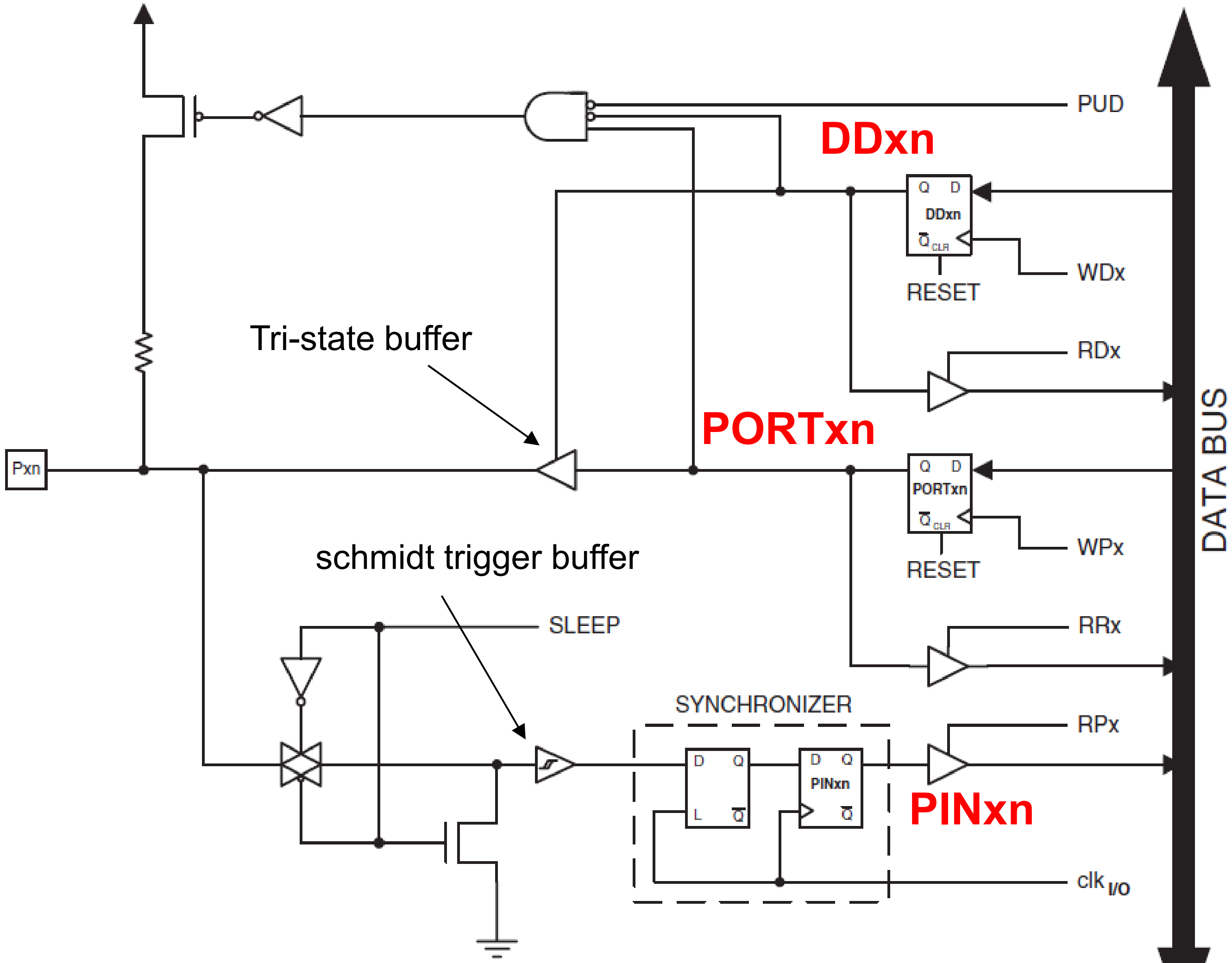
Data Register – PORTx ➔ PORTxn
Data Direction Register – DDRx ➔ DDxn
Port Input Pins – PINx ➔ PINxn

ATmega128 Port Architecture

- **Port: bi-directional I/O ports with optional internal pull-ups**



General Digital I/O



DDxn

PORTxn

PINxn

Tri-state buffer

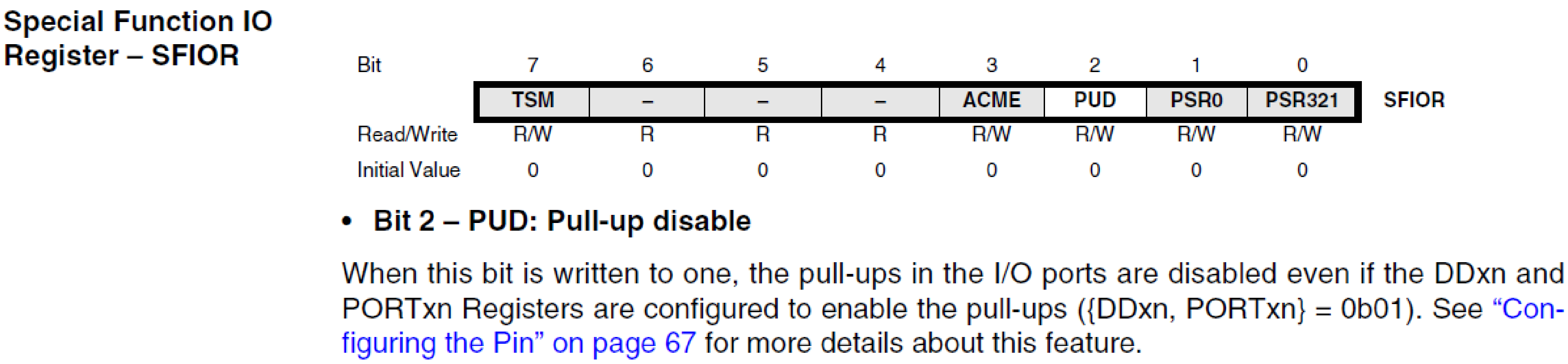
schmidt trigger buffer

PUD:	PULLUP DISABLE
SLEEP:	SLEEP CONTROL
$clk_{I/O}$:	I/O CLOCK

WDx:	WRITE DDRx
RDx:	READ DDRx
WPx:	WRITE PORTx
RRx:	READ PORTx REGISTER
RPx:	READ PORTx PIN

Port Pin Configurations

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)



Special Function IO Register – SFIOR

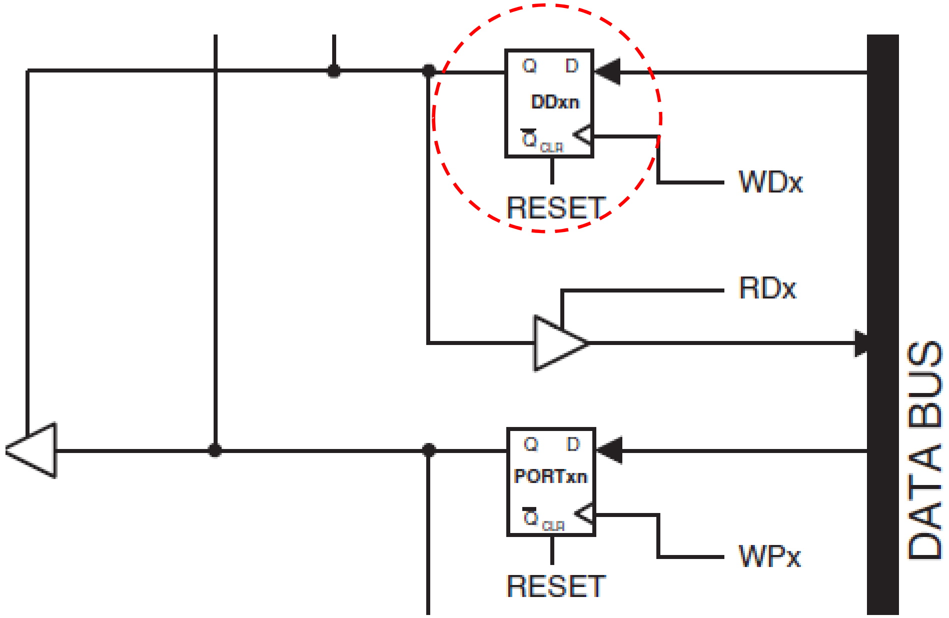
Bit	7	6	5	4	3	2	1	0	
	TSM	–	–	–	ACME	PUD	PSR0	PSR321	SFIOR
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 2 – PUD: Pull-up disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01). See "Configuring the Pin" on page 67 for more details about this feature.

Configuring the Pin - DDxn

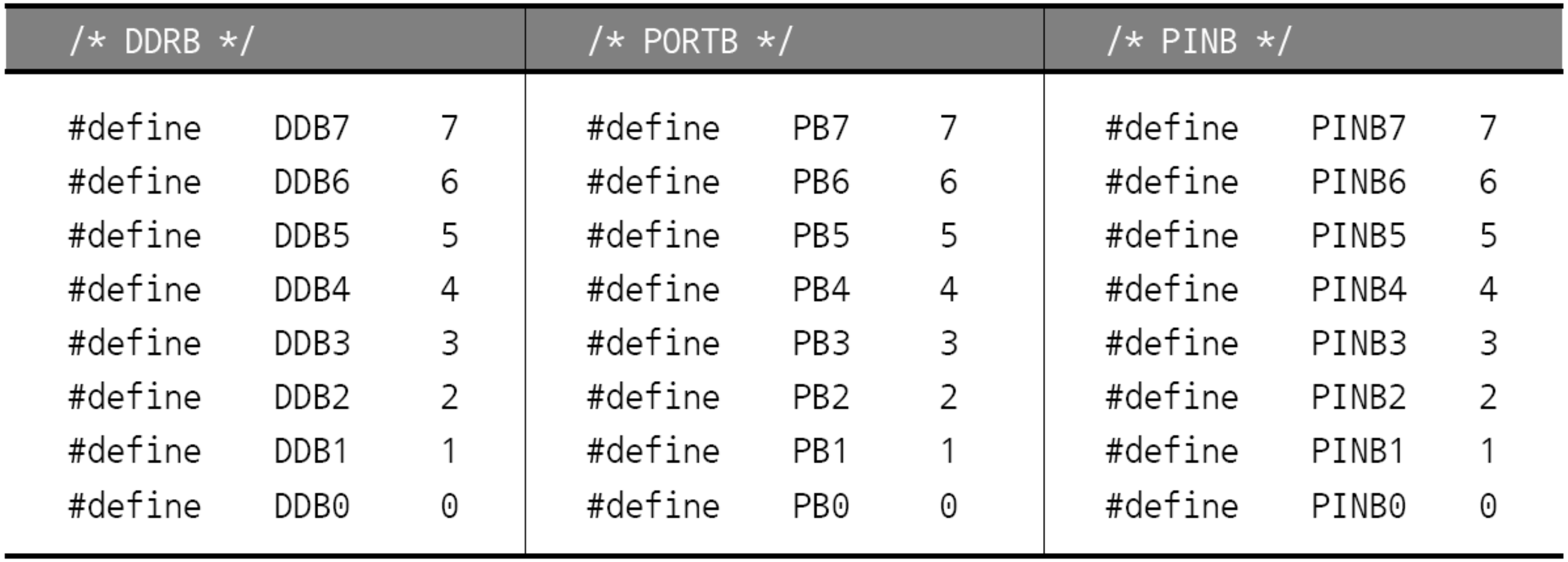
- **DDxn bit in the DDRx Register**
  - Selects the direction of the pin
  - 1: output, 0: input



Bitwise Operation Example

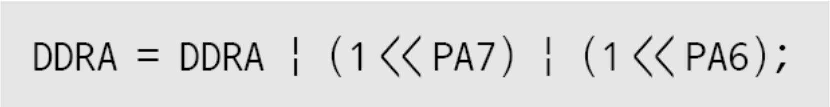
#include <avr/io.h>

[B포트의 DDRB, PORTB, PINB 레지스터와 비트명]



/* DDRB */	/* PORTB */	/* PINB */
#define DDB7 7	#define PB7 7	#define PINB7 7
#define DDB6 6	#define PB6 6	#define PINB6 6
#define DDB5 5	#define PB5 5	#define PINB5 5
#define DDB4 4	#define PB4 4	#define PINB4 4
#define DDB3 3	#define PB3 3	#define PINB3 3
#define DDB2 2	#define PB2 2	#define PINB2 2
#define DDB1 1	#define PB1 1	#define PINB1 1
#define DDB0 0	#define PB0 0	#define PINB0 0

Bitwise Operation Example



```
DDRA = DDRA | (1<<PA7) | (1<<PA6);
```

		7	6	5	4	3	2	1	0
❶	1<< PA7	1	0	0	0	0	0	0	0

		7	6	5	4	3	2	1	0
❷	1<< PA6	0	1	0	0	0	0	0	0

Bitwise Operation Example

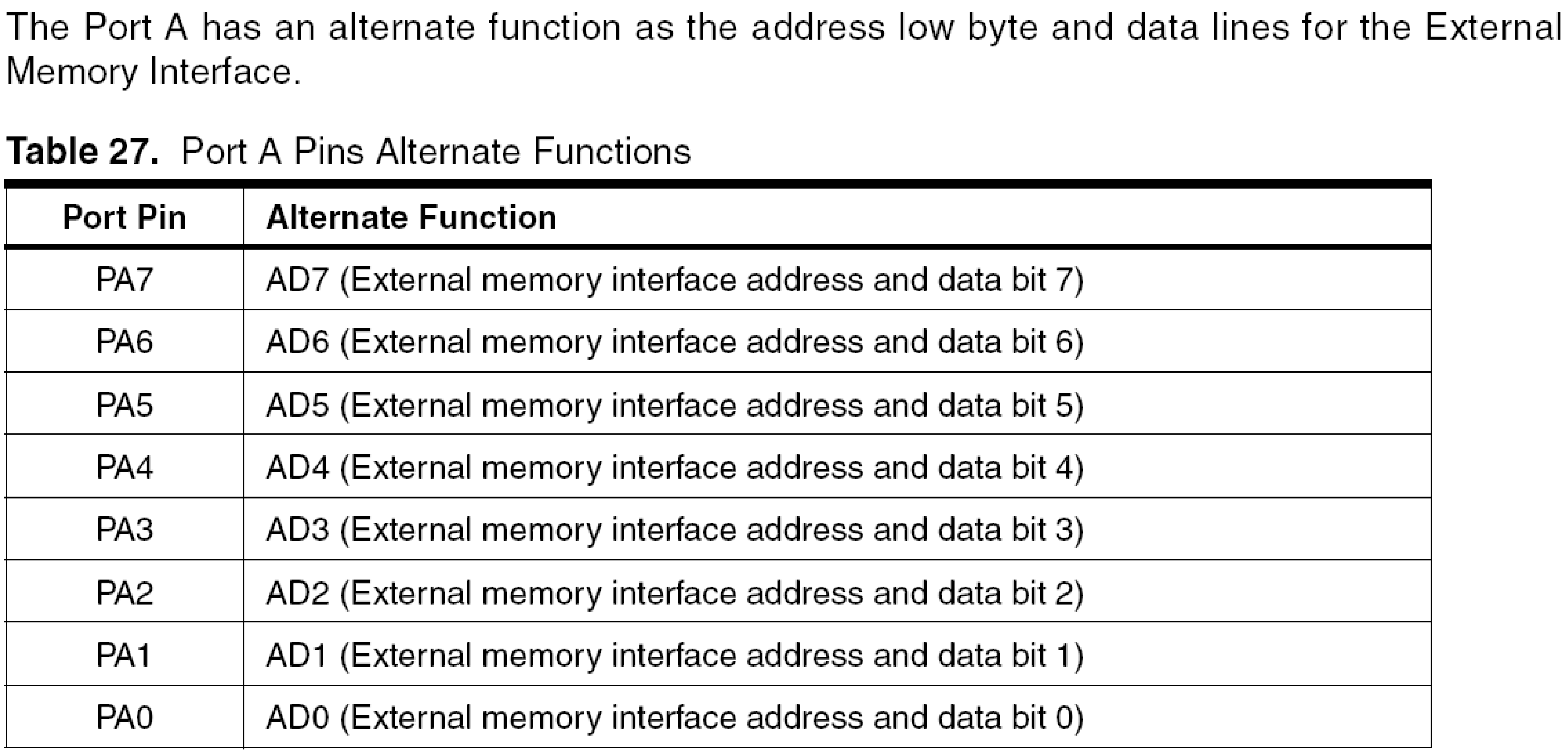
❸

	7	6	5	4	3	2	1	0
DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0
1<< PA7	1	0	0	0	0	0	0	0
DDRA | (1<< PA7)	1	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0

❹

	7	6	5	4	3	2	1	0
DDRA | (1<< PA7)	1	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0
1<< PA6	0	1	0	0	0	0	0	0
DDRA | (1<< PA7) | (1<< PA6)	1	1	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0

Alternate Functions of Port A



The Port A has an alternate function as the address low byte and data lines for the External Memory Interface.

Table 27. Port A Pins Alternate Functions

Port Pin	Alternate Function
PA7	AD7 (External memory interface address and data bit 7)
PA6	AD6 (External memory interface address and data bit 6)
PA5	AD5 (External memory interface address and data bit 5)
PA4	AD4 (External memory interface address and data bit 4)
PA3	AD3 (External memory interface address and data bit 3)
PA2	AD2 (External memory interface address and data bit 2)
PA1	AD1 (External memory interface address and data bit 1)
PA0	AD0 (External memory interface address and data bit 0)

Alternate Functions of Port B

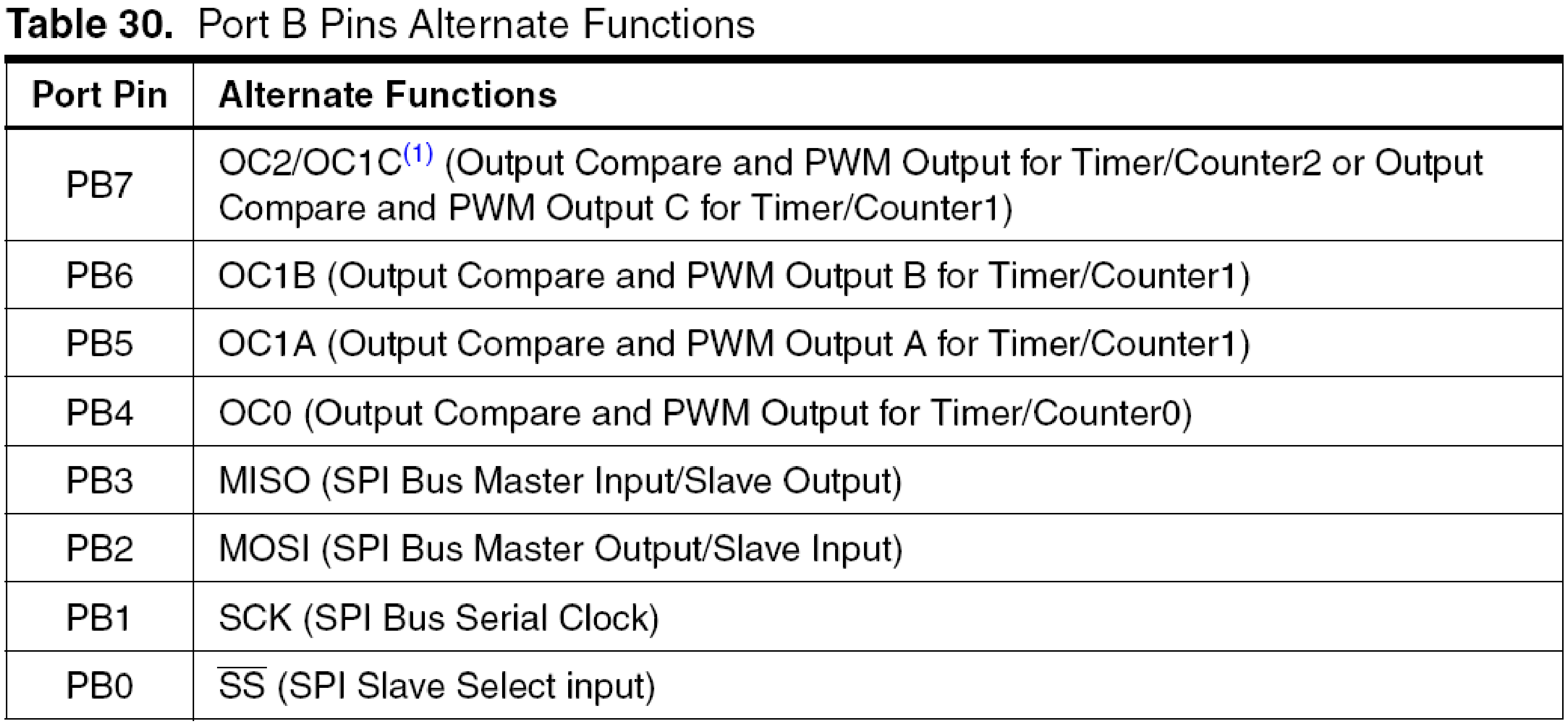
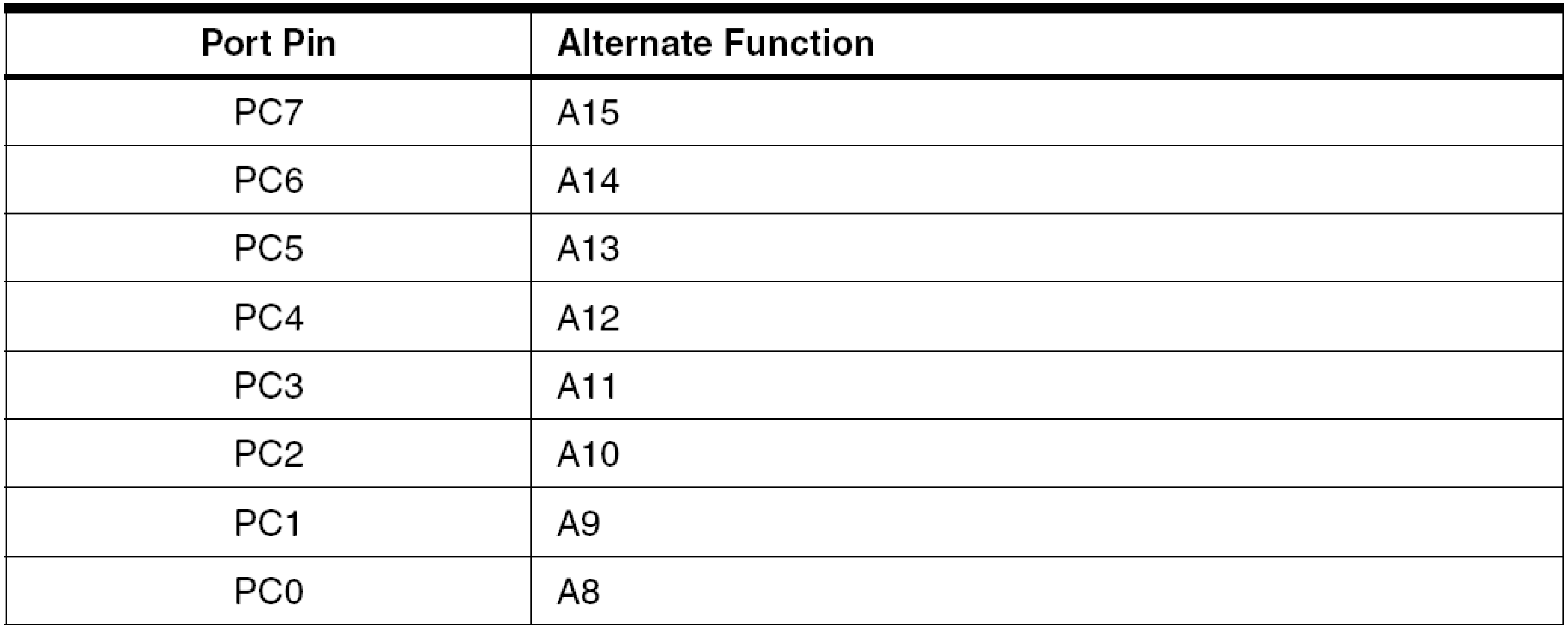


Table 30. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	OC2/OC1C(1) (Output Compare and PWM Output for Timer/Counter2 or Output Compare and PWM Output C for Timer/Counter1)
PB6	OC1B (Output Compare and PWM Output B for Timer/Counter1)
PB5	OC1A (Output Compare and PWM Output A for Timer/Counter1)
PB4	OC0 (Output Compare and PWM Output for Timer/Counter0)
PB3	MISO (SPI Bus Master Input/Slave Output)
PB2	MOSI (SPI Bus Master Output/Slave Input)
PB1	SCK (SPI Bus Serial Clock)
PB0	$\overline{\text{SS}}$ (SPI Slave Select input)

Alternate Functions of Port C

Table 33. Port C Pins Alternate Functions



Port Pin	Alternate Function
PC7	A15
PC6	A14
PC5	A13
PC4	A12
PC3	A11
PC2	A10
PC1	A9
PC0	A8

Alternate Functions of Port D

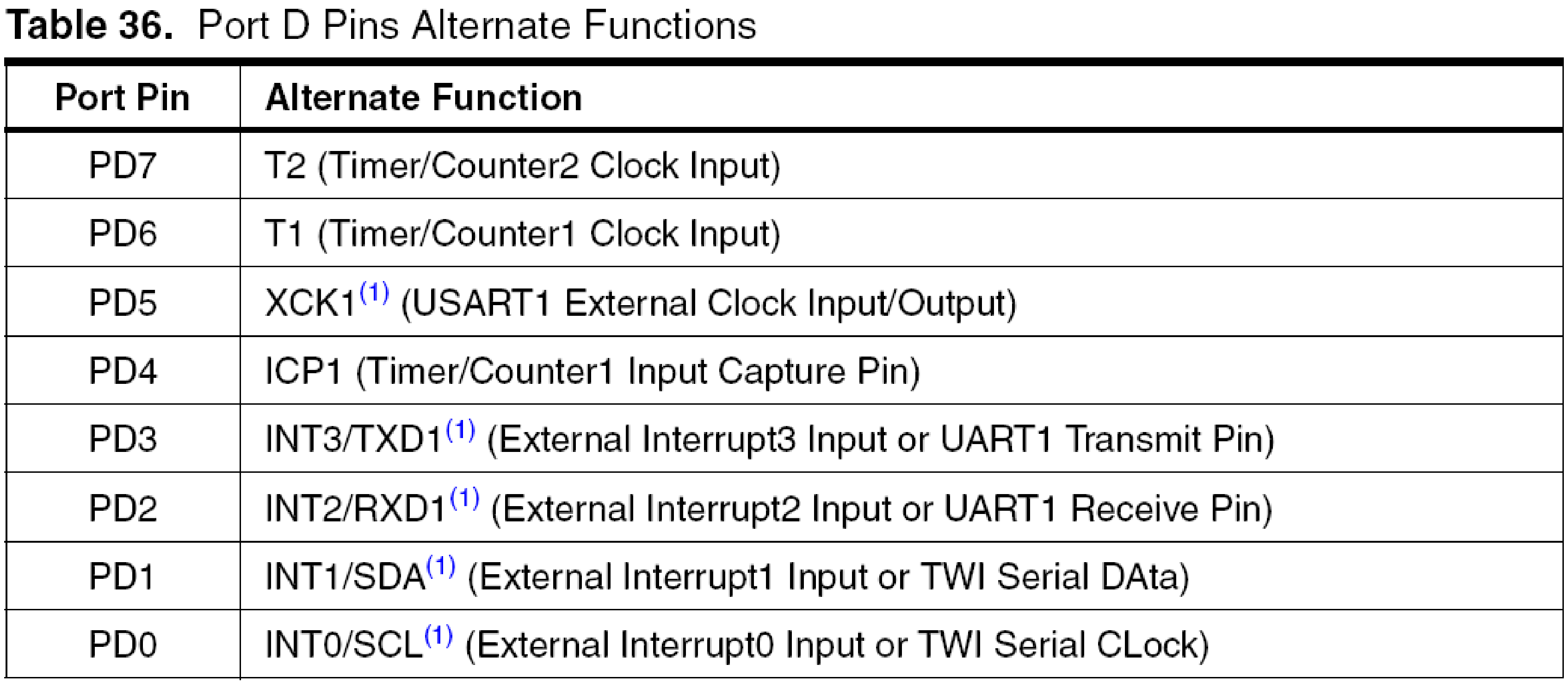
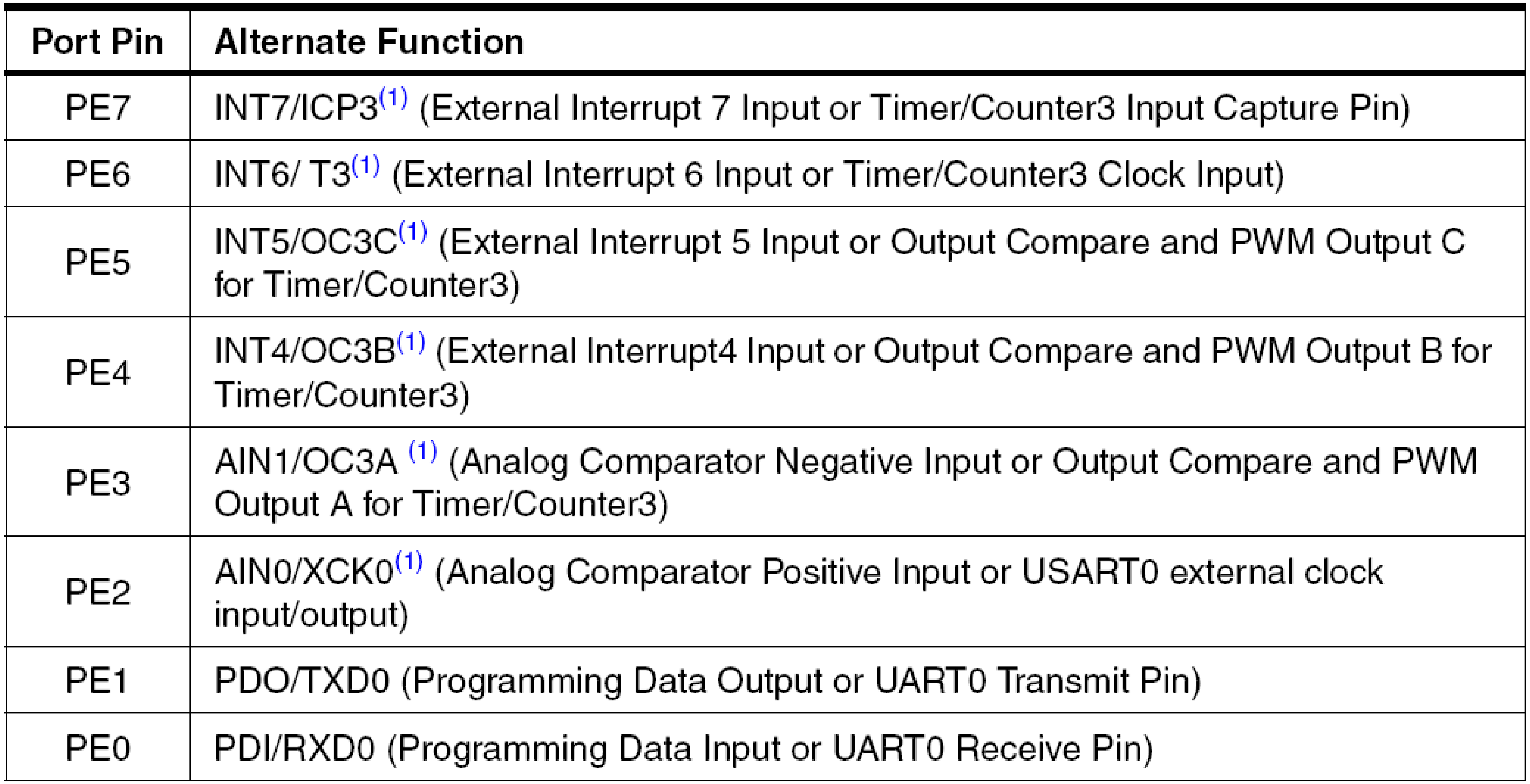


Table 36. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD7	T2 (Timer/Counter2 Clock Input)
PD6	T1 (Timer/Counter1 Clock Input)
PD5	XCK1[1] (USART1 External Clock Input/Output)
PD4	ICP1 (Timer/Counter1 Input Capture Pin)
PD3	INT3/TXD1[1] (External Interrupt3 Input or UART1 Transmit Pin)
PD2	INT2/RXD1[1] (External Interrupt2 Input or UART1 Receive Pin)
PD1	INT1/SDA[1] (External Interrupt1 Input or TWI Serial DAta)
PD0	INT0/SCL[1] (External Interrupt0 Input or TWI Serial CLock)

Alternate Functions of Port E

Table 39. Port E Pins Alternate Functions



Port Pin	Alternate Function
PE7	INT7/ICP3[1] (External Interrupt 7 Input or Timer/Counter3 Input Capture Pin)
PE6	INT6/ T3[1] (External Interrupt 6 Input or Timer/Counter3 Clock Input)
PE5	INT5/OC3C[1] (External Interrupt 5 Input or Output Compare and PWM Output C for Timer/Counter3)
PE4	INT4/OC3B[1] (External Interrupt4 Input or Output Compare and PWM Output B for Timer/Counter3)
PE3	AIN1/OC3A [1] (Analog Comparator Negative Input or Output Compare and PWM Output A for Timer/Counter3)
PE2	AIN0/XCK0[1] (Analog Comparator Positive Input or USART0 external clock input/output)
PE1	PDO/TXD0 (Programming Data Output or UART0 Transmit Pin)
PE0	PDI/RXD0 (Programming Data Input or UART0 Receive Pin)

Alternate Functions of Port F

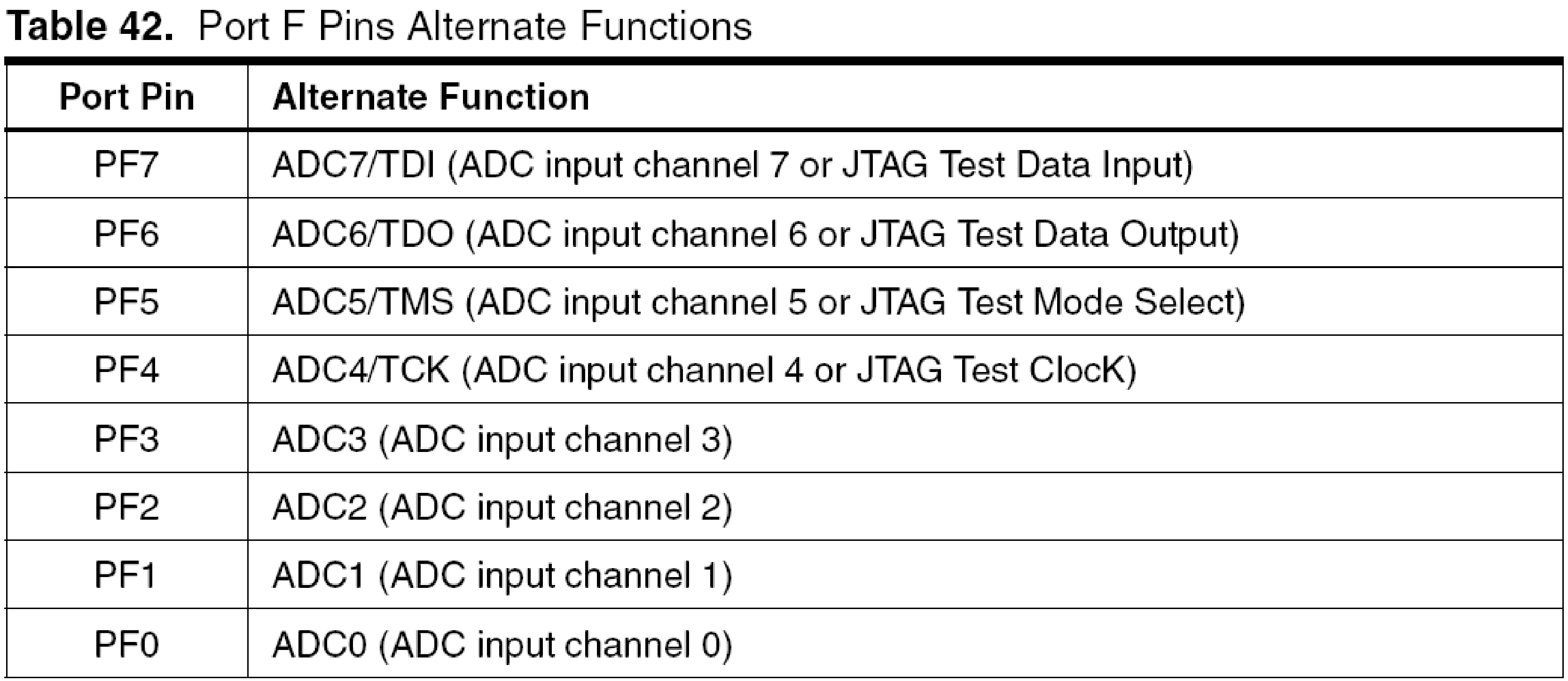


Table 42. Port F Pins Alternate Functions

Port Pin	Alternate Function
PF7	ADC7/TDI (ADC input channel 7 or JTAG Test Data Input)
PF6	ADC6/TDO (ADC input channel 6 or JTAG Test Data Output)
PF5	ADC5/TMS (ADC input channel 5 or JTAG Test Mode Select)
PF4	ADC4/TCK (ADC input channel 4 or JTAG Test ClocK)
PF3	ADC3 (ADC input channel 3)
PF2	ADC2 (ADC input channel 2)
PF1	ADC1 (ADC input channel 1)
PF0	ADC0 (ADC input channel 0)

Alternate Functions of Port G

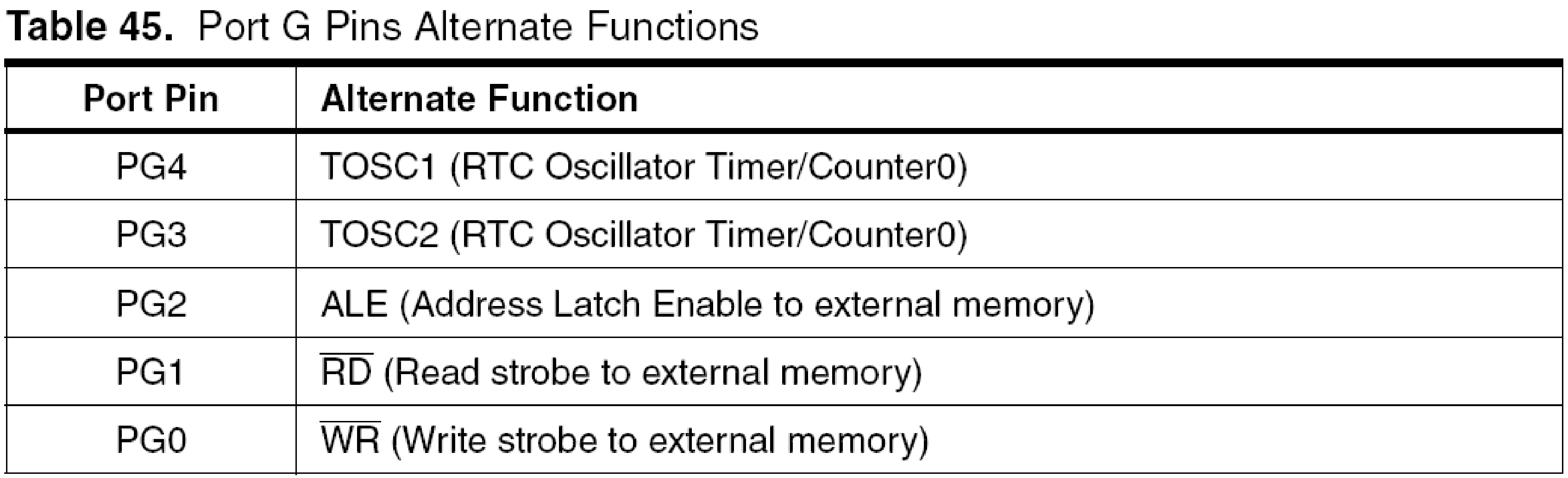
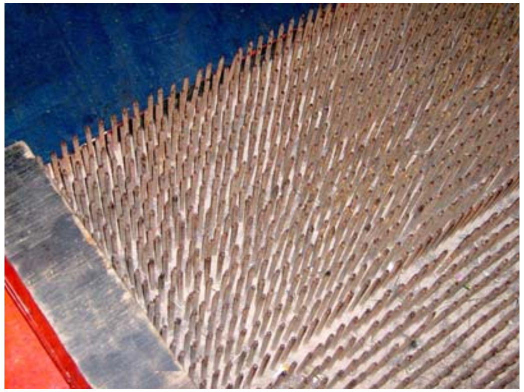


Table 45. Port G Pins Alternate Functions

Port Pin	Alternate Function
PG4	TOSC1 (RTC Oscillator Timer/Counter0)
PG3	TOSC2 (RTC Oscillator Timer/Counter0)
PG2	ALE (Address Latch Enable to external memory)
PG1	$\overline{\text{RD}}$ (Read strobe to external memory)
PG0	$\overline{\text{WR}}$ (Write strobe to external memory)

JTAG/IEEE1149.1

- 배경
  - 초기에는 조립된 PCB(Printed Circuit Board)의 시험을 목적으로 연구
  - 기존의 테스트: Bed-of-Nails Probing
  - 디바이스 패키징 및 부품 실장 기술 발전
    - SMD 출현, 핀의 고집적화(예:BGA)
    - 고집적, 표면 실장, 양면 실장, 다층화
  - Physical access가 어렵게 되어 기존 방법으로 테스트가 어려움
  - Programmable logic device의 on-board programming이 요구됨



- 해결책
  - 테스트를 위해 표준화된 로직(TAP) 및 테스트 핀 (Boundary Scan Cell)을 칩에 내장
  - 테스트 방법 표준화
    - Boundary Scan Architecture
    - Test Access Port(TAP)

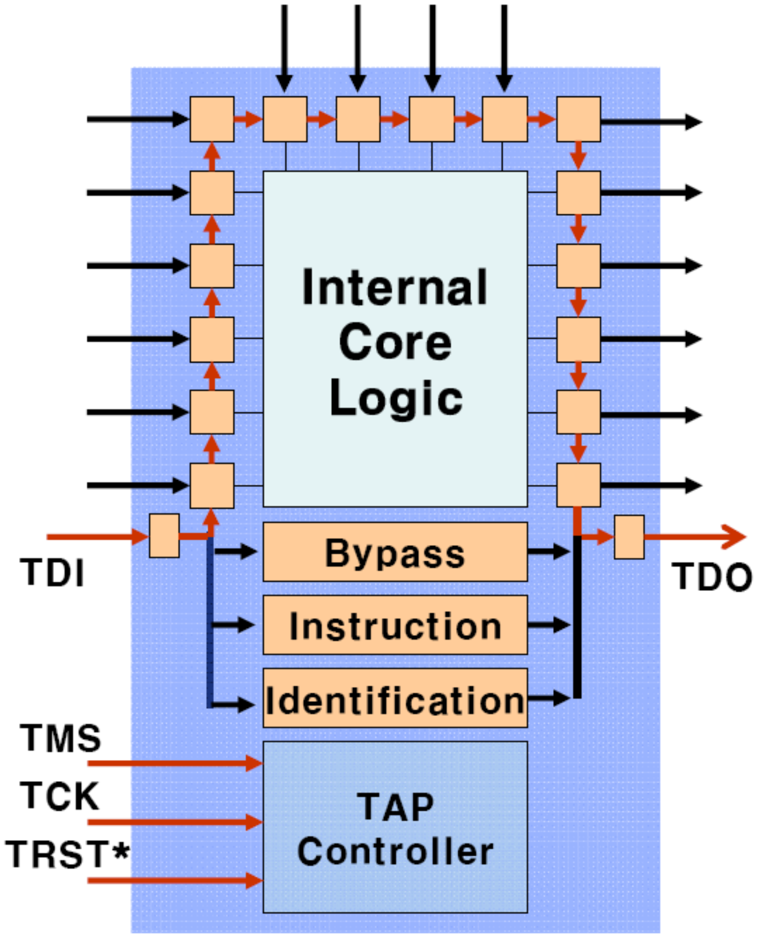
JTAG/IEEE1149.1

- 많은 프로세서 제조업체가 JTAG interface를 확장 및 응용
  - On-chip debug feature (e.g. on-chip breakpoints)에 접근 및 제어
  - DRAM 등의 메모리 접근 또는 플래시 등의 로직 디바이스 프로그래밍

- JTAG interface는 시리얼 인터페이스를 가지며 기본적으로 다음 5개의 signal로 구성
  - nTRST: test reset
  - TMS: test mode select
  - TDI: test data input
  - TDO: test data output
  - TCK: test clock

- 디버그 목적으로 몇 개의 신호를 추가
  - nRESET(=nSRST): chip reset
  - VTref: 기준 전압 제공
  - DBGRQ, DBGACK: 외부 trigger signal을 사용

JTAG/IEEE1149.1



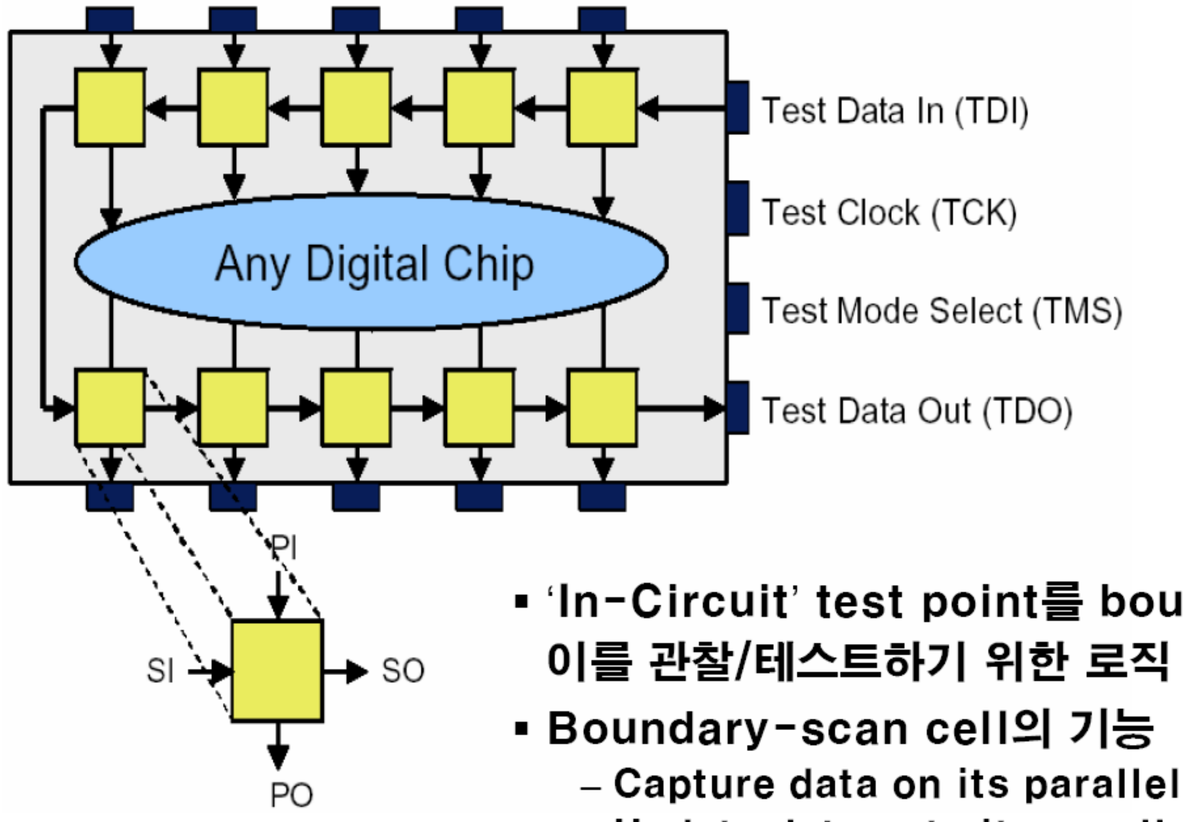
◆ IEEE 1149.1 Device Architecture

- 칩 내부에 본래의 기능을 구현한 Core Logic과 테스트 및 기타 목적을 위한 JTAG 블록이 있다

- JTAG 블록은 TAP Controller와 Register들로 구성되어 있으며 TDI, TDO를 통해 데이터를 주고 받을 수 있다

- JTAG operation은 TAP controller에 의해 제어된다

JTAG: Boundary Scan



Boundary Scan Cell

- ‘In-Circuit’ test point를 boundary로 옮기고 이를 관찰/테스트하기 위한 로직 추가
- Boundary-scan cell의 기능
  - Capture data on its parallel input PI
  - Update data onto its parallel output PO
  - Serially scan data from SO to its neighbor's SI
  - Behave transparently: PI passes to PO