



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

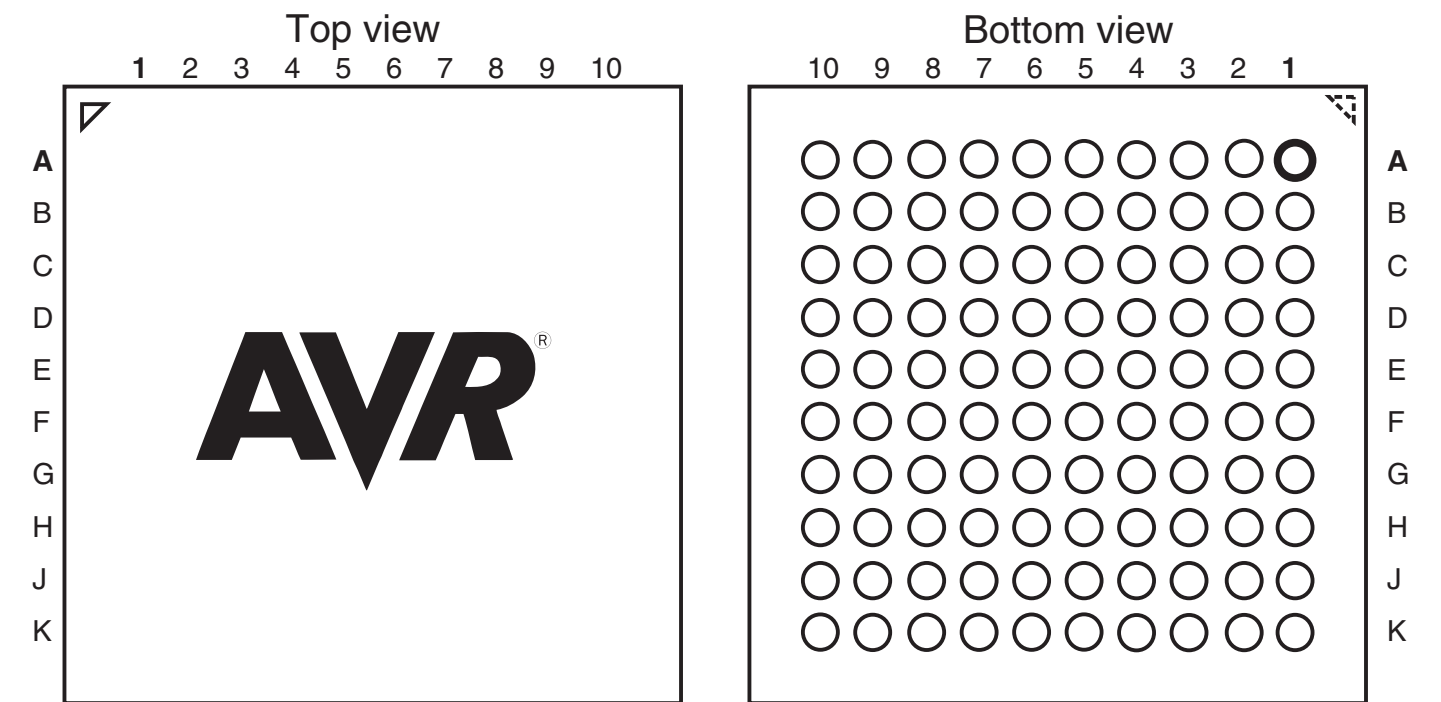
"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	8MHz
Connectivity	EBI/EMI, I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	86
Program Memory Size	256KB (128K x 16)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/atmega2560v-8ai">https://www.e-xfl.com/product-detail/microchip-technology/atmega2560v-8ai</a>

**Figure 1-2.** CBGA-pinout ATmega640/1280/2560

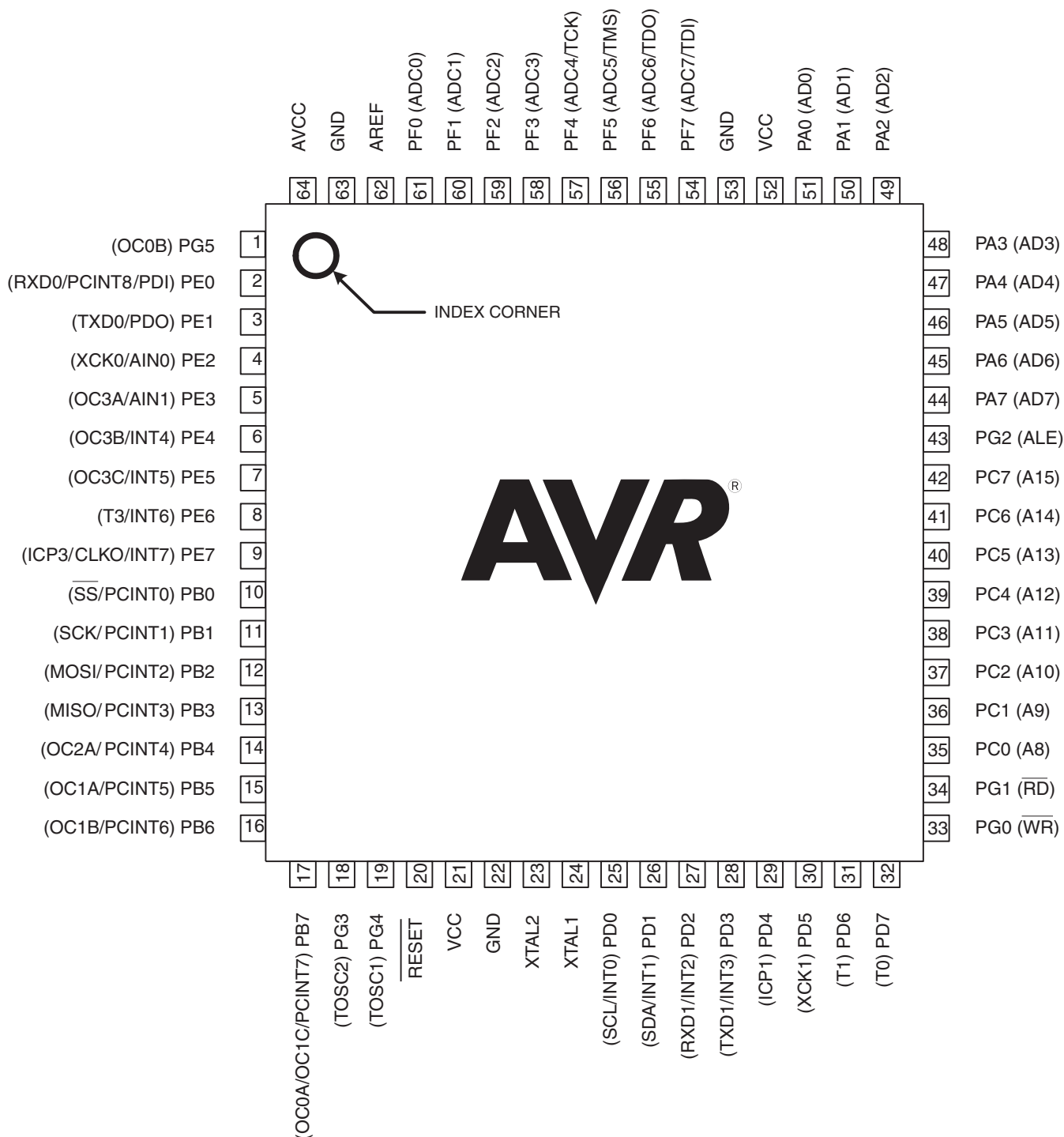


**Table 1-1.** CBGA-pinout ATmega640/1280/2560

	1	2	3	4	5	6	7	8	9	10
A	GND	AREF	PF0	PF2	PF5	PK0	PK3	PK6	GND	VCC
B	AVCC	PG5	PF1	PF3	PF6	PK1	PK4	PK7	PA0	PA2
C	PE2	PE0	PE1	PF4	PF7	PK2	PK5	PJ7	PA1	PA3
D	PE3	PE4	PE5	PE6	PH2	PA4	PA5	PA6	PA7	PG2
E	PE7	PH0	PH1	PH3	PH5	PJ6	PJ5	PJ4	PJ3	PJ2
F	VCC	PH4	PH6	PB0	PL4	PD1	PJ1	PJ0	PC7	GND
G	GND	PB1	PB2	PB5	PL2	PD0	PD5	PC5	PC6	VCC
H	PB3	PB4	RESET	PL1	PL3	PL7	PD4	PC4	PC3	PC2
J	PH7	PG3	PB6	PL0	XTAL2	PL6	PD3	PC1	PC0	PG1
K	PB7	PG4	VCC	GND	XTAL1	PL5	PD2	PD6	PD7	PG0

Note: The functions for each pin is the same as for the 100 pin packages shown in [Figure 1-1 on page 2](#).

**Figure 1-3.** Pinout ATmega1281/2561



**Note:** The large center pad underneath the QFN/MLF package is made of metal and internally connected to GND. It should be soldered or glued to the board to ensure good mechanical stability. If the center pad is left unconnected, the package might loosen from the board.

The ATmega640/1280/1281/2560/2561 provides the following features: 64K/128K/256K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4Kbytes EEPROM, 8Kbytes SRAM, 54/86 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), six flexible Timer/Counters with compare modes and PWM, four USARTs, a byte oriented 2-wire Serial Interface, a 16-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE® std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS®) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using the Atmel high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega640/1280/1281/2560/2561 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega640/1280/1281/2560/2561 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

### 2.3.6 Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on [page 80](#).

### 2.3.7 Port E (PE7..PE0)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on [page 82](#).

### 2.3.8 Port F (PF7..PF0)

Port F serves as analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a reset occurs.

Port F also serves the functions of the JTAG interface.

### 2.3.9 Port G (PG5..PG0)

Port G is a 6-bit I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on [page 86](#).

### 2.3.10 Port H (PH7..PH0)

Port H is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port H output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port H pins that are externally pulled low will source current if the pull-up resistors are activated. The Port H pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port H also serves the functions of various special features of the ATmega640/1280/2560 as listed on [page 88](#).

### 2.3.11 Port J (PJ7..PJ0)

Port J is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port J output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port J pins that are externally pulled low will source current if the pull-up resistors are activated. The Port J pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port J also serves the functions of various special features of the ATmega640/1280/2560 as listed on [page 90](#).

### 3. Resources

A comprehensive set of development tools and application notes, and datasheets are available for download on <http://www.atmel.com/avr>.

### 4. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Confirm with the C compiler documentation for more details.

These code examples assume that the part specific header file is included before compilation. For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBR", "SBRC", "SBR", and "CBR".

### 5. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 ppm over 20 years at 85°C or 100 years at 25°C.

### 6. Capacitive touch sensing

The Atmel® QTouch® Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR® microcontrollers. The QTouch Library includes support for the QTouch and QMatrix acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

The QTouch Library is FREE and downloadable from the Atmel website at the following location: [www.atmel.com/qtouchlibrary](http://www.atmel.com/qtouchlibrary). For implementation details and other information, refer to the [Atmel QTouch Library User Guide](#) - also available for download from the Atmel website.

## 7. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page	
(0x1FF)	Reserved	-	-	-	-	-	-	-	-		
...	Reserved	-	-	-	-	-	-	-	-		
(0x13F)	Reserved										
(0x13E)	Reserved										
(0x13D)	Reserved										
(0x13C)	Reserved										
(0x13B)	Reserved										
(0x13A)	Reserved										
(0x139)	Reserved										
(0x138)	Reserved										
(0x137)	Reserved										
(0x136)	UDR3	USART3 I/O Data Register								<a href="#">page 218</a>	
(0x135)	UBRR3H	-	-	-	-	USART3 Baud Rate Register High Byte					<a href="#">page 222</a>
(0x134)	UBRR3L	USART3 Baud Rate Register Low Byte								<a href="#">page 222</a>	
(0x133)	Reserved	-	-	-	-	-	-	-	-		
(0x132)	UCSR3C	UMSEL31	UMSEL30	UPM31	UPM30	USBS3	UCSZ31	UCSZ30	UCPOL3	<a href="#">page 235</a>	
(0x131)	UCSR3B	RXCIE3	TXCIE3	UDRIE3	RXEN3	TXEN3	UCSZ32	RXB83	TXB83	<a href="#">page 234</a>	
(0x130)	UCSR3A	RXC3	TXC3	UDRE3	FE3	DOR3	UPE3	U2X3	MPCM3	<a href="#">page 233</a>	
(0x12F)	Reserved	-	-	-	-	-	-	-	-		
(0x12E)	Reserved	-	-	-	-	-	-	-	-		
(0x12D)	OCR5CH	Timer/Counter5 - Output Compare Register C High Byte								<a href="#">page 160</a>	
(0x12C)	OCR5CL	Timer/Counter5 - Output Compare Register C Low Byte								<a href="#">page 160</a>	
(0x12B)	OCR5BH	Timer/Counter5 - Output Compare Register B High Byte								<a href="#">page 160</a>	
(0x12A)	OCR5BL	Timer/Counter5 - Output Compare Register B Low Byte								<a href="#">page 160</a>	
(0x129)	OCR5AH	Timer/Counter5 - Output Compare Register A High Byte								<a href="#">page 160</a>	
(0x128)	OCR5AL	Timer/Counter5 - Output Compare Register A Low Byte								<a href="#">page 160</a>	
(0x127)	ICR5H	Timer/Counter5 - Input Capture Register High Byte								<a href="#">page 161</a>	
(0x126)	ICR5L	Timer/Counter5 - Input Capture Register Low Byte								<a href="#">page 161</a>	
(0x125)	TCNT5H	Timer/Counter5 - Counter Register High Byte								<a href="#">page 158</a>	
(0x124)	TCNT5L	Timer/Counter5 - Counter Register Low Byte								<a href="#">page 158</a>	
(0x123)	Reserved	-	-	-	-	-	-	-	-		
(0x122)	TCCR5C	FOC5A	FOC5B	FOC5C	-	-	-	-	-	<a href="#">page 157</a>	
(0x121)	TCCR5B	ICNC5	ICES5	-	WGM53	WGM52	CS52	CS51	CS50	<a href="#">page 156</a>	
(0x120)	TCCR5A	COM5A1	COM5A0	COM5B1	COM5B0	COM5C1	COM5C0	WGM51	WGM50	<a href="#">page 154</a>	
(0x11F)	Reserved	-	-	-	-	-	-	-	-		
(0x11E)	Reserved	-	-	-	-	-	-	-	-		
(0x11D)	Reserved	-	-	-	-	-	-	-	-		
(0x11C)	Reserved	-	-	-	-	-	-	-	-		
(0x11B)	Reserved	-	-	-	-	-	-	-	-		
(0x11A)	Reserved	-	-	-	-	-	-	-	-		
(0x119)	Reserved	-	-	-	-	-	-	-	-		
(0x118)	Reserved	-	-	-	-	-	-	-	-		
(0x117)	Reserved	-	-	-	-	-	-	-	-		
(0x116)	Reserved	-	-	-	-	-	-	-	-		
(0x115)	Reserved	-	-	-	-	-	-	-	-		
(0x114)	Reserved	-	-	-	-	-	-	-	-		
(0x113)	Reserved	-	-	-	-	-	-	-	-		
(0x112)	Reserved	-	-	-	-	-	-	-	-		
(0x111)	Reserved	-	-	-	-	-	-	-	-		
(0x110)	Reserved	-	-	-	-	-	-	-	-		
(0x10F)	Reserved	-	-	-	-	-	-	-	-		
(0x10E)	Reserved	-	-	-	-	-	-	-	-		
(0x10D)	Reserved	-	-	-	-	-	-	-	-		
(0x10C)	Reserved	-	-	-	-	-	-	-	-		
(0x10B)	PORTL	PORTL7	PORTL6	PORTL5	PORTL4	PORTL3	PORTL2	PORTL1	PORTL0	<a href="#">page 100</a>	
(0x10A)	DDRL	DDL7	DDL6	DDL5	DDL4	DDL3	DDL2	DDL1	DDL0	<a href="#">page 100</a>	
(0x109)	PINL	PINL7	PINL6	PINL5	PINL4	PINL3	PINL2	PINL1	PINL0	<a href="#">page 100</a>	
(0x108)	PORTK	PORTK7	PORTK6	PORTK5	PORTK4	PORTK3	PORTK2	PORTK1	PORTK0	<a href="#">page 99</a>	
(0x107)	DDRK	DDK7	DDK6	DDK5	DDK4	DDK3	DDK2	DDK1	DDK0	<a href="#">page 99</a>	
(0x106)	PINK	PINK7	PINK6	PINK5	PINK4	PINK3	PINK2	PINK1	PINK0	<a href="#">page 99</a>	
(0x105)	PORTJ	PORTJ7	PORTJ6	PORTJ5	PORTJ4	PORTJ3	PORTJ2	PORTJ1	PORTJ0	<a href="#">page 99</a>	
(0x104)	DDRJ	DDJ7	DDJ6	DDJ5	DDJ4	DDJ3	DDJ2	DDJ1	DDJ0	<a href="#">page 99</a>	
(0x103)	PINJ	PINJ7	PINJ6	PINJ5	PINJ4	PINJ3	PINJ2	PINJ1	PINJ0	<a href="#">page 99</a>	
(0x102)	PORTH	PORTH7	PORTH6	PORTH5	PORTH4	PORTH3	PORTH2	PORTH1	PORTH0	<a href="#">page 98</a>	
(0x101)	DDRH	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0	<a href="#">page 99</a>	

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x100)	PINH	PINH7	PINH6	PINH5	PINH4	PINH3	PINH2	PINH1	PINH0	<a href="#">page 99</a>
(0xFF)	Reserved	-	-	-	-	-	-	-	-	
(0xFE)	Reserved	-	-	-	-	-	-	-	-	
(0xFD)	Reserved	-	-	-	-	-	-	-	-	
(0xFC)	Reserved	-	-	-	-	-	-	-	-	
(0xFB)	Reserved	-	-	-	-	-	-	-	-	
(0xFA)	Reserved	-	-	-	-	-	-	-	-	
(0xF9)	Reserved	-	-	-	-	-	-	-	-	
(0xF8)	Reserved	-	-	-	-	-	-	-	-	
(0xF7)	Reserved	-	-	-	-	-	-	-	-	
(0xF6)	Reserved	-	-	-	-	-	-	-	-	
(0xF5)	Reserved	-	-	-	-	-	-	-	-	
(0xF4)	Reserved	-	-	-	-	-	-	-	-	
(0xF3)	Reserved	-	-	-	-	-	-	-	-	
(0xF2)	Reserved	-	-	-	-	-	-	-	-	
(0xF1)	Reserved	-	-	-	-	-	-	-	-	
(0xF0)	Reserved	-	-	-	-	-	-	-	-	
(0xEF)	Reserved	-	-	-	-	-	-	-	-	
(0xEE)	Reserved	-	-	-	-	-	-	-	-	
(0xED)	Reserved	-	-	-	-	-	-	-	-	
(0xEC)	Reserved	-	-	-	-	-	-	-	-	
(0xEB)	Reserved	-	-	-	-	-	-	-	-	
(0xEA)	Reserved	-	-	-	-	-	-	-	-	
(0xE9)	Reserved	-	-	-	-	-	-	-	-	
(0xE8)	Reserved	-	-	-	-	-	-	-	-	
(0xE7)	Reserved	-	-	-	-	-	-	-	-	
(0xE6)	Reserved	-	-	-	-	-	-	-	-	
(0xE5)	Reserved	-	-	-	-	-	-	-	-	
(0xE4)	Reserved	-	-	-	-	-	-	-	-	
(0xE3)	Reserved	-	-	-	-	-	-	-	-	
(0xE2)	Reserved	-	-	-	-	-	-	-	-	
(0xE1)	Reserved	-	-	-	-	-	-	-	-	
(0xE0)	Reserved	-	-	-	-	-	-	-	-	
(0xDF)	Reserved	-	-	-	-	-	-	-	-	
(0xDE)	Reserved	-	-	-	-	-	-	-	-	
(0xDD)	Reserved	-	-	-	-	-	-	-	-	
(0xDC)	Reserved	-	-	-	-	-	-	-	-	
(0xDB)	Reserved	-	-	-	-	-	-	-	-	
(0xDA)	Reserved	-	-	-	-	-	-	-	-	
(0xD9)	Reserved	-	-	-	-	-	-	-	-	
(0xD8)	Reserved	-	-	-	-	-	-	-	-	
(0xD7)	Reserved	-	-	-	-	-	-	-	-	
(0xD6)	UDR2	USART2 I/O Data Register								<a href="#">page 218</a>
(0xD5)	UBRR2H	-	-	-	-	USART2 Baud Rate Register High Byte				<a href="#">page 222</a>
(0xD4)	UBRR2L	USART2 Baud Rate Register Low Byte								<a href="#">page 222</a>
(0xD3)	Reserved	-	-	-	-	-	-	-	-	
(0xD2)	UCSR2C	UMSEL21	UMSEL20	UPM21	UPM20	USBS2	UCSZ21	UCSZ20	UCPOL2	<a href="#">page 235</a>
(0xD1)	UCSR2B	RXCIE2	TXCIE2	UDRIE2	RXEN2	TXEN2	UCSZ22	RXB82	TXB82	<a href="#">page 234</a>
(0xD0)	UCSR2A	RXC2	TXC2	UDRE2	FE2	DOR2	UPE2	U2X2	MPCM2	<a href="#">page 233</a>
(0xCF)	Reserved	-	-	-	-	-	-	-	-	
(0xCE)	UDR1	USART1 I/O Data Register								<a href="#">page 218</a>
(0xCD)	UBRR1H	-	-	-	-	USART1 Baud Rate Register High Byte				<a href="#">page 222</a>
(0xCC)	UBRR1L	USART1 Baud Rate Register Low Byte								<a href="#">page 222</a>
(0xCB)	Reserved	-	-	-	-	-	-	-	-	
(0xCA)	UCSR1C	UMSEL11	UMSEL10	UPM11	UPM10	USBS1	UCSZ11	UCSZ10	UCPOL1	<a href="#">page 235</a>
(0xC9)	UCSR1B	RXCIE1	TXCIE1	UDRIE1	RXEN1	TXEN1	UCSZ12	RXB81	TXB81	<a href="#">page 234</a>
(0xC8)	UCSR1A	RXC1	TXC1	UDRE1	FE1	DOR1	UPE1	U2X1	MPCM1	<a href="#">page 233</a>
(0xC7)	Reserved	-	-	-	-	-	-	-	-	
(0xC6)	UDR0	USART0 I/O Data Register								<a href="#">page 218</a>
(0xC5)	UBRR0H	-	-	-	-	USART0 Baud Rate Register High Byte				<a href="#">page 222</a>
(0xC4)	UBRR0L	USART0 Baud Rate Register Low Byte								<a href="#">page 222</a>
(0xC3)	Reserved	-	-	-	-	-	-	-	-	
(0xC2)	UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0	<a href="#">page 235</a>
(0xC1)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	<a href="#">page 234</a>
(0xC0)	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	<a href="#">page 234</a>
(0xBF)	Reserved	-	-	-	-	-	-	-	-	
(0xBE)	Reserved	-	-	-	-	-	-	-	-	
(0xBD)	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	-	<a href="#">page 264</a>



Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x14 (0x34)	PORTG	-	-	PORTG5	PORTG4	PORTG3	PORTG2	PORTG1	PORTG0	<a href="#">page 98</a>
0x13 (0x33)	DDRG	-	-	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	<a href="#">page 98</a>
0x12 (0x32)	PING	-	-	PING5	PING4	PING3	PING2	PING1	PING0	<a href="#">page 98</a>
0x11 (0x31)	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	<a href="#">page 97</a>
0x10 (0x30)	DDRF	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	<a href="#">page 98</a>
0x0F (0x2F)	PINF	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	<a href="#">page 98</a>
0x0E (0x2E)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	<a href="#">page 97</a>
0x0D (0x2D)	DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	<a href="#">page 97</a>
0x0C (0x2C)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	<a href="#">page 98</a>
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	<a href="#">page 97</a>
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	<a href="#">page 97</a>
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	<a href="#">page 97</a>
0x08 (0x28)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	<a href="#">page 97</a>
0x07 (0x27)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	<a href="#">page 97</a>
0x06 (0x26)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	<a href="#">page 97</a>
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	<a href="#">page 96</a>
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	<a href="#">page 96</a>
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	<a href="#">page 96</a>
0x02 (0x22)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	<a href="#">page 96</a>
0x01 (0x21)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	<a href="#">page 96</a>
0x00 (0x20)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	<a href="#">page 96</a>

- Notes:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
  2. I/O registers within the address range \$00 - \$1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
  3. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
  4. When using the I/O specific commands IN and OUT, the I/O addresses \$00 - \$3F must be used. When addressing I/O registers as data space using LD and ST instructions, \$20 must be added to these addresses. The ATmega640/1280/1281/2560/2561 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from \$60 - \$1FF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

## 8. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
<b>ARITHMETIC AND LOGIC INSTRUCTIONS</b>					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z, C, N, V, H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z, C, N, V, H	1
ADIW	RdI, K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z, C, N, V, S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z, C, N, V, H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z, C, N, V, H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z, C, N, V, H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z, C, N, V, H	1
SBIW	RdI, K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z, C, N, V, S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \bullet Rr$	Z, N, V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \bullet K$	Z, N, V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z, N, V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z, N, V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z, N, V	1
COM	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z, C, N, V	1
NEG	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z, C, N, V, H	1
SBR	Rd, K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z, N, V	1
CBR	Rd, K	Clear Bit(s) in Register	$Rd \leftarrow Rd \bullet (0xFF - K)$	Z, N, V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z, N, V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z, N, V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \bullet Rd$	Z, N, V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z, N, V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z, C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z, C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z, C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z, C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z, C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z, C	2
<b>BRANCH INSTRUCTIONS</b>					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
JMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
EIJMP		Extended Indirect Jump to (Z)	$PC \leftarrow (EIND:Z)$	None	2
JMP	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	4
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	4
EICALL		Extended Indirect Call to (Z)	$PC \leftarrow (EIND:Z)$	None	4
CALL	k	Direct Subroutine Call	$PC \leftarrow k$	None	5
RET		Subroutine Return	$PC \leftarrow STACK$	None	5
RETI		Interrupt Return	$PC \leftarrow STACK$	I	5
CPSE	Rd, Rr	Compare, Skip if Equal	if $(Rd = Rr)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
CP	Rd, Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
CPC	Rd, Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
CPI	Rd, K	Compare Register with Immediate	$Rd - K$	Z, N, V, C, H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if $(Rr(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	if $(Rr(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if $(P(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	if $(P(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if $(SREG(s) = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if $(SREG(s) = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BREQ	k	Branch if Equal	if $(Z = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Branch if Not Equal	if $(Z = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Branch if Carry Set	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Branch if Carry Cleared	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Branch if Same or Higher	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Branch if Lower	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Branch if Minus	if $(N = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Branch if Plus	if $(N = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if $(H = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if $(H = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	if $(T = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	if $(T = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if $(V = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2

Mnemonics	Operands	Description	Operation	Flags	#Clocks
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
<b>MCU CONTROL INSTRUCTIONS</b>					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

Note: EICALL and EIJMP do not exist in ATmega640/1280/1281.  
ELPM does not exist in ATmega640.

## 9. Ordering Information

### 9.1 ATmega640

Speed [MHz] <sup>(2)</sup>	Power Supply	Ordering Code	Package <sup>(1)(3)</sup>	Operation Range
8	1.8 - 5.5V	ATmega640V-8AU ATmega640V-8AUR <sup>(4)</sup> ATmega640V-8CU ATmega640V-8CUR <sup>(4)</sup>	100A 100A 100C1 100C1	Industrial (-40°C to 85°C)
16	2.7 - 5.5V	ATmega640-16AU ATmega640-16AUR <sup>(4)</sup> ATmega640-16CU ATmega640-16CUR <sup>(4)</sup>	100A 100A 100C1 100C1	

- Notes:
1. This device can also be supplied in wafer form. Contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. See [“Speed Grades” on page 357](#).
  3. Pb-free packaging, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  4. Tape & Reel.

Package Type	
<b>100A</b>	100-lead, Thin (1.0mm) Plastic Gull Wing Quad Flat Package (TQFP)
<b>100C1</b>	100-ball, Chip Ball Grid Array (CBGA)

## 9.2 ATmega1280

Speed [MHz] <sup>(2)</sup>	Power Supply	Ordering Code	Package <sup>(1)(3)</sup>	Operation Range
8	1.8V - 5.5V	ATmega1280V-8AU	100A	Industrial (-40°C to 85°C)
		ATmega1280V-8AUR <sup>(4)</sup>	100A	
		ATmega1280V-8CU	100C1	
		ATmega1280V-8CUR <sup>(4)</sup>	100C1	
16	2.7V - 5.5V	ATmega1280-16AU	100A	
		ATmega1280-16AUR <sup>(4)</sup>	100A	
		ATmega1280-16CU	100C1	
		ATmega1280-16CUR <sup>(4)</sup>	100C1	

- Notes:
1. This device can also be supplied in wafer form. Contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. See [“Speed Grades” on page 357](#).
  3. Pb-free packaging, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  4. Tape & Reel.

Package Type	
<b>100A</b>	100-lead, Thin (1.0mm) Plastic Gull Wing Quad Flat Package (TQFP)
<b>100C1</b>	100-ball, Chip Ball Grid Array (CBGA)

## 9.5 ATmega2561

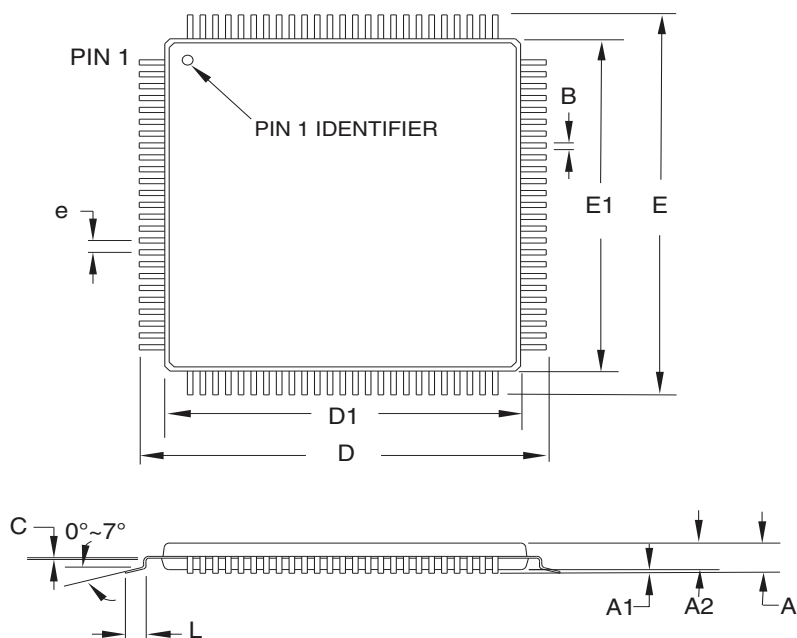
Speed [MHz] <sup>(2)</sup>	Power Supply	Ordering Code	Package <sup>(1)(3)</sup>	Operation Range
8	1.8V - 5.5V	ATmega2561V-8AU ATmega2561V-8AUR <sup>(4)</sup> ATmega2561V-8MU ATmega2561V-8MUR <sup>(4)</sup>	64A 64A 64M2 64M2	Industrial (-40°C to 85°C)
16	4.5V - 5.5V	ATmega2561-16AU ATmega2561-16AUR <sup>(4)</sup> ATmega2561-16MU ATmega2561-16MUR <sup>(4)</sup>	64A 64A 64M2 64M2	

- Notes:
1. This device can also be supplied in wafer form. Contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. See [“Speed Grades” on page 357](#).
  3. Pb-free packaging, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  4. Tape & Reel.

Package Type	
<b>64A</b>	64-lead, Thin (1.0mm) Plastic Gull Wing Quad Flat Package (TQFP)
<b>64M2</b>	64-pad, 9mm × 9mm × 1.0mm Body, Quad Flat No-lead/Micro Lead Frame Package (QFN/MLF)

## 10. Packaging Information

### 10.1 100A



#### COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	—	—	1.20	
A1	0.05	—	0.15	
A2	0.95	1.00	1.05	
D	15.75	16.00	16.25	
D1	13.90	14.00	14.10	Note 2
E	15.75	16.00	16.25	
E1	13.90	14.00	14.10	Note 2
B	0.17	—	0.27	
C	0.09	—	0.20	
L	0.45	—	0.75	
e	0.50 TYP			

#### Notes:

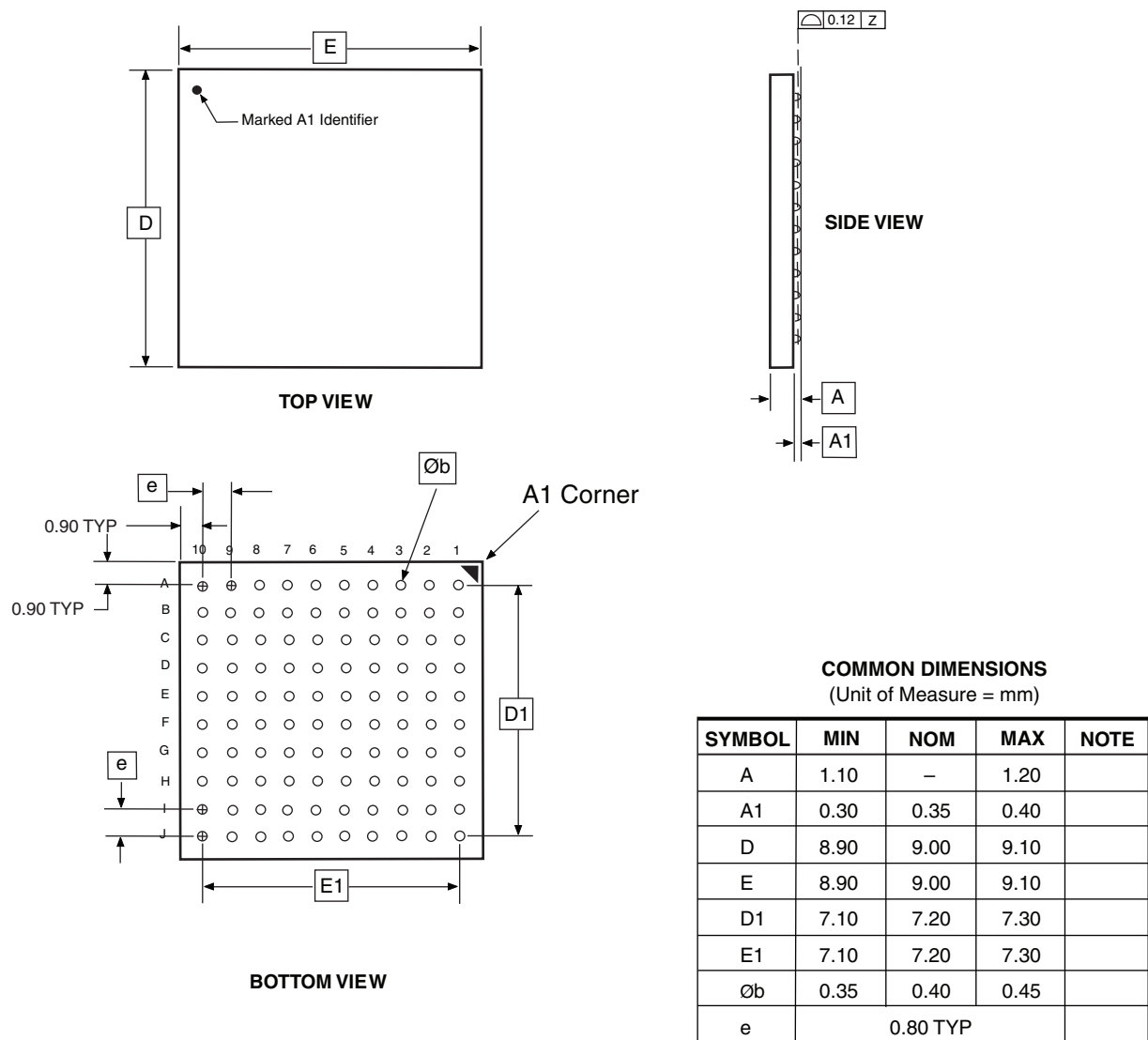
1. This package conforms to JEDEC reference MS-026, Variation AED.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.08 mm maximum.

2010-10-20

TITLE		DRAWING NO.	REV.
<b>Atmel</b> Package Drawing Contact: <a href="mailto:packagedrawings@atmel.com">packagedrawings@atmel.com</a>		100A	D

**100A**, 100-lead, 14 x 14 mm Body Size, 1.0 mm Body Thickness,  
0.5 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)

10.2 100C1



5/25/06



2325 Orchard Parkway  
San Jose, CA 95131

**TITLE**

**100C1**, 100-ball, 9 x 9 x 1.2 mm Body, Ball Pitch 0.80 mm  
Chip Array BGA Package (CBGA)

**DRAWING NO.**

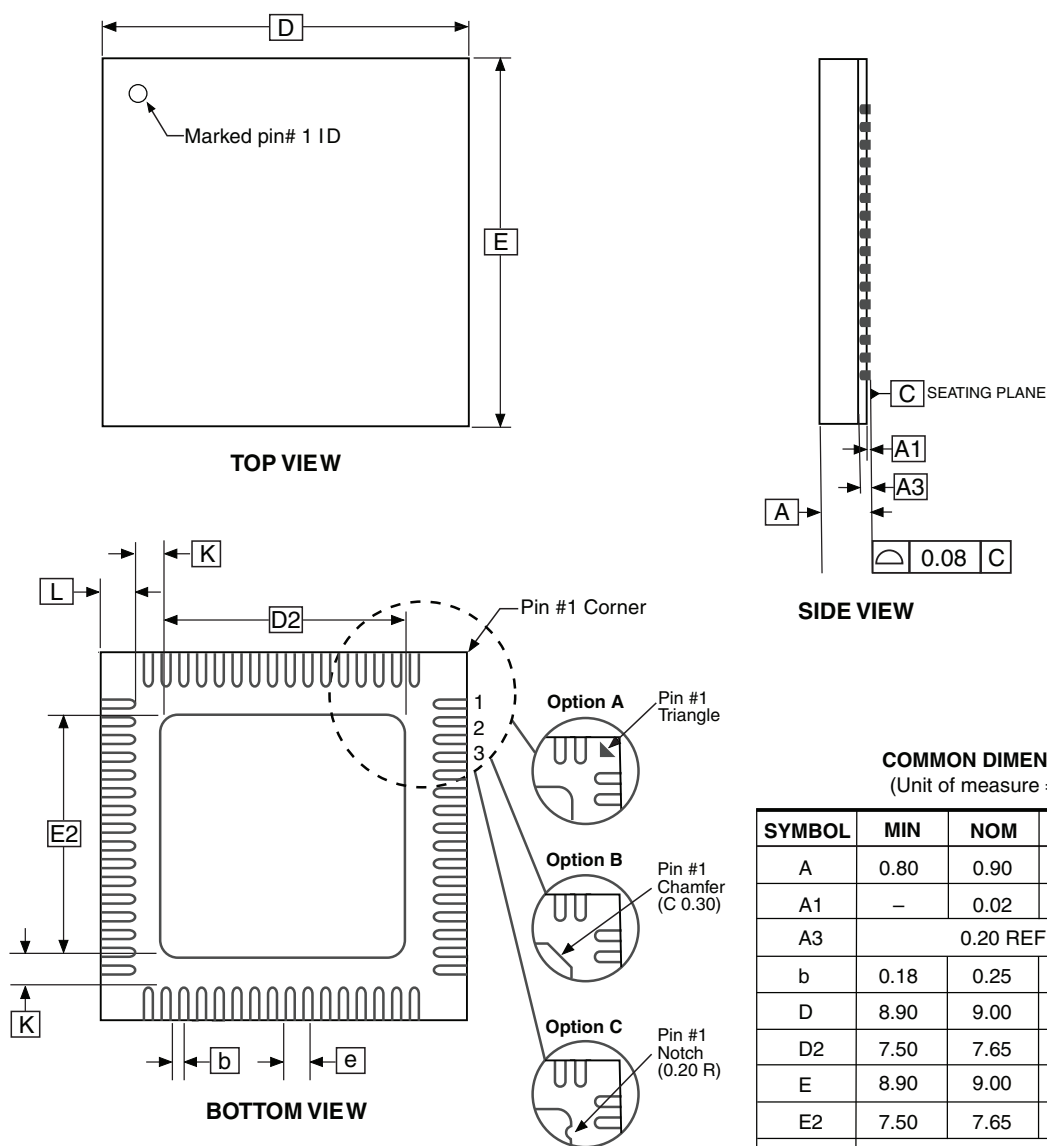
100C1

**REV.**

A



## 10.4 64M2



**COMMON DIMENSIONS**  
(Unit of measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	—	0.02	0.05	
A3	0.20 REF			
b	0.18	0.25	0.30	
D	8.90	9.00	9.10	
D2	7.50	7.65	7.80	
E	8.90	9.00	9.10	
E2	7.50	7.65	7.80	
e	0.50 BSC			
L	0.35	0.40	0.45	
K	0.20	0.27	0.40	

Notes: 1. JEDEC Standard MO-220, (SAW Singulation) fig. 1, VMMD.  
2. Dimension and tolerance conform to ASMEY14.5M-1994.

2014-02-12

**Atmel** 2325 Orchard Parkway  
San Jose, CA 95131

### TITLE

**64M2**, 64-pad, 9 x 9 x 1.0mm Body, Lead Pitch 0.50mm,  
7.65mm Exposed Pad, Micro Lead Frame Package (MLF)

### DRAWING NO.

64M2

### REV.

E

**Problem Fix/Workaround**

None.

**2. High current consumption in sleep mode**

If a pending interrupt cannot wake the part up from the selected sleep mode, the current consumption will increase during sleep when executing the SLEEP instruction directly after a SEI instruction.

**Problem Fix/Workaround**

Before entering sleep, interrupts not used to wake the part from the sleep mode should be disabled.

**11.5 ATmega1281 rev. B**

- **High current consumption in sleep mode**

**1. High current consumption in sleep mode**

If a pending interrupt cannot wake the part up from the selected sleep mode, the current consumption will increase during sleep when executing the SLEEP instruction directly after a SEI instruction.

**Problem Fix/Workaround**

Before entering sleep, interrupts not used to wake the part from the sleep mode should be disabled.

**11.6 ATmega1281 rev. A**

- **Inaccurate ADC conversion in differential mode with 200× gain**
- **High current consumption in sleep mode**

**1. Inaccurate ADC conversion in differential mode with 200× gain**

With AVCC <3.6V, random conversions will be inaccurate. Typical absolute accuracy may reach 64 LSB.

**Problem Fix/Workaround**

None.

**2. High current consumption in sleep mode**

If a pending interrupt cannot wake the part up from the selected sleep mode, the current consumption will increase during sleep when executing the SLEEP instruction directly after a SEI instruction.

**Problem Fix/Workaround**

Before entering sleep, interrupts not used to wake the part from the sleep mode should be disabled.

**11.7 ATmega2560 rev. F**

- **ADC differential input amplification by 46dB (200x) not functional**

**1. ADC differential input amplification by 46dB (200x) not functional****Problem Fix/Workaround**

None.

**11.8 ATmega2560 rev. E**

No known errata.

**11.9 ATmega2560 rev. D**

Not sampled.

## 11.10 ATmega2560 rev. C

- High current consumption in sleep mode

### 1. High current consumption in sleep mode

If a pending interrupt cannot wake the part up from the selected sleep mode, the current consumption will increase during sleep when executing the SLEEP instruction directly after a SEI instruction.

#### Problem Fix/Workaround

Before entering sleep, interrupts not used to wake the part from the sleep mode should be disabled.

## 11.11 ATmega2560 rev. B

Not sampled.

## 11.12 ATmega2560 rev. A

- Non-Read-While-Write area of flash not functional
- Part does not work under 2.4 volts
- Incorrect ADC reading in differential mode
- Internal ADC reference has too low value
- IN/OUT instructions may be executed twice when Stack is in external RAM
- EEPROM read from application code does not work in Lock Bit Mode 3

### 1. Non-Read-While-Write area of flash not functional

The Non-Read-While-Write area of the flash is not working as expected. The problem is related to the speed of the part when reading the flash of this area.

#### Problem Fix/Workaround

- Only use the first 248K of the flash.

- If boot functionality is needed, run the code in the Non-Read-While-Write area at maximum 1/4th of the maximum frequency of the device at any given voltage. This is done by writing the CLKPR register before entering the boot section of the code.

### 2. Part does not work under 2.4 volts

The part does not execute code correctly below 2.4 volts.

#### Problem Fix/Workaround

Do not use the part at voltages below 2.4 volts.

### 3. Incorrect ADC reading in differential mode

The ADC has high noise in differential mode. It can give up to 7 LSB error.

#### Problem Fix/Workaround

Use only the 7 MSB of the result when using the ADC in differential mode.

### 4. Internal ADC reference has too low value

The internal ADC reference has a value lower than specified.

#### Problem Fix/Workaround

- Use AVCC or external reference.

- The actual value of the reference can be measured by applying a known voltage to the ADC when using the internal reference. The result when doing later conversions can then be calibrated.

## 5. IN/OUT instructions may be executed twice when Stack is in external RAM

If either an IN or an OUT instruction is executed directly before an interrupt occurs and the stack pointer is located in external ram, the instruction will be executed twice. In some cases this will cause a problem, for example:

- If reading SREG it will appear that the I-flag is cleared.
- If writing to the PIN registers, the port will toggle twice.
- If reading registers with interrupt flags, the flags will appear to be cleared.

### Problem Fix/Workaround

There are two application workarounds, where selecting one of them, will be omitting the issue:

- Replace IN and OUT with LD/LDS/LDD and ST/STS/STD instructions.
- Use internal RAM for stack pointer.

## 6. EEPROM read from application code does not work in Lock Bit Mode 3

When the Memory Lock Bits LB2 and LB1 are programmed to mode 3, EEPROM read does not work from the application code.

### Problem Fix/Workaround

Do not set Lock Bit Protection Mode 3 when the application code needs to read from EEPROM.

## 11.13 ATmega2561 rev. F

- ADC differential input amplification by 46dB (200x) not functional

### 1. ADC differential input amplification by 46dB (200x) not functional

#### Problem Fix/Workaround

None.

## 11.14 ATmega2561 rev. E

No known errata.

## 11.15 ATmega2561 rev. D

Not sampled.

## 11.16 ATmega2561 rev. C

- High current consumption in sleep mode.

### 1. High current consumption in sleep mode

If a pending interrupt cannot wake the part up from the selected sleep mode, the current consumption will increase during sleep when executing the SLEEP instruction directly after a SEI instruction.

#### Problem Fix/Workaround

Before entering sleep, interrupts not used to wake the part from the sleep mode should be disabled.

## 11.17 ATmega2561 rev. B

Not sampled.

