Welcome to 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	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	CANbus, Ethernet, I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	81
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 16x12b; D/A 4x12b
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	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3c1512c-aut

The Full-Speed USB 2.0 Device interface supports several USB Classes at the same time thanks to the rich End-Point configuration. The On-The-GO (OTG) Host interface allows device like a USB Flash disk or a USB printer to be directly connected to the processor.

The media-independent interface (MII) and reduced MII (RMII) 10/100 Ethernet MAC module provides on-chip solutions for network-connected devices.

The Peripheral Event Controller (PEVC) allows to redirect events from one peripheral or from input pins to another peripheral. It can then trigger, in a deterministic time, an action inside a peripheral without the need of CPU. For instance a PWM waveform can directly trigger an ADC capture, hence avoiding delays due to software interrupt processing.

The AT32UC3C features analog functions like ADC, DAC, Analog comparators. The ADC interface is built around a 12-bit pipelined ADC core and is able to control two independent 8-channel or one 16-channel. The ADC block is able to measure two different voltages sampled at the same time. The analog comparators can be paired to detect when the sensing voltage is within or outside the defined reference window.

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 included 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.

AT32UC3C integrates a class 2+ Nexus 2.0 On-Chip Debug (OCD) System, with non-intrusive real-time trace, full-speed read/write memory access in addition to basic runtime control. The Nanotrace interface enables trace feature for aWire- or JTAG-based debuggers. The single-pin aWire interface allows all features available through the JTAG interface to be accessed through the RESET pin, allowing the JTAG pins to be used for GPIO or peripherals.

2.2 Configuration Summary

Table 2-1. Configuration Summary

Feature	AT32UC3C0512C/ AT32UC3C0256C/ AT32UC3C0128C/ AT32UC3C064C	AT32UC3C1512C/ AT32UC3C1256C/ AT32UC3C1128C/ AT32UC3C164C	AT32UC3C2512C/ AT32UC3C2256C/ AT32UC3C2128C/ AT32UC3C264C
Flash	512/256/128/64 KB	512/256/128/64 KB	512/256/128/64 KB
SRAM	64/64/32/16 KB	64/64/32/16 KB	64/64/32/16 KB
HSB RAM	4 KB		
EBI	1	0	0
GPIO	123	81	45
External Interrupts	8	8	8
TWI	3	3	2
USART	5	5	4
Peripheral DMA Channels	16	16	16
Peripheral Event System	1	1	1
SPI	2	2	1
CAN channels	2	2	2
USB	1	1	1
Ethernet MAC 10/100	1 RMII/MII	1 RMII/MII	1 RMII only
I2S	1	1	1
Asynchronous Timers	1	1	1
Timer/Counter Channels	6	6	3
PWM channels	4x2		
QDEC	2	2	1
Frequency Meter	1		
Watchdog Timer	1		
Power Manager	1		
Oscillators	PLL 80-240 MHz (PLL0/PLL1) Crystal Oscillator 0.4-20 MHz (OSC0) Crystal Oscillator 32 KHz (OSC32K) RC Oscillator 115 kHz (RCSYS) RC Oscillator 8 MHz (RC8M) RC Oscillator 120 MHz (RC120M)		
	0.4-20 MHz (OSC1)		-
12-bit ADC number of channels	1 16	1 16	1 11
12-bit DAC number of channels	1 4	1 4	1 2

Table 2-1. Configuration Summary

Feature	AT32UC3C0512C/ AT32UC3C0256C/ AT32UC3C0128C/ AT32UC3C064C	AT32UC3C1512C/ AT32UC3C1256C/ AT32UC3C1128C/ AT32UC3C164C	AT32UC3C2512C/ AT32UC3C2256C/ AT32UC3C2128C/ AT32UC3C264C
Analog Comparators	4	4	2
JTAG	1		
aWire	1		
Max Frequency	66 MHz		
Package	LQFP144	TQFP100	TQFP64/QFN64

Table 3-2. Peripheral Functions

Function	Description
aWire DATAOUT	aWire output in two-pin mode
JTAG port connections	JTAG debug port
Oscillators	OSC0, OSC32

3.2.3 Oscillator Pinout

The oscillators are not mapped to the normal GPIO functions and their muxings are controlled by registers in the System Control Interface (SCIF). Please refer to the SCIF chapter for more information about this.

Table 3-3. Oscillator pinout

QFN64/ TQFP64 pin	TQFP100 pin	LQFP144 pin	Pad	Oscillator pin
31	47	69	PB30	xin0
	99	143	PB02	xin1
62	96	140	PB00	xin32
32	48	70	PB31	xout0
	100	144	PB03	xout1
63	97	141	PB01	xout32

3.2.4 JTAG port connections

If the JTAG is enabled, the JTAG will take control over a number of pins, irrespectively of the I/O Controller configuration.

Table 3-4. JTAG pinout

QFN64/ TQFP64 pin	TQFP100 pin	LQFP144 pin	Pin name	JTAG pin
2	2	2	PA01	TDI
3	3	3	PA02	TDO
4	4	4	PA03	TMS
1	1	1	PA00	TCK

3.2.5 Nexus OCD AUX port connections

If the OCD trace system is enabled, the trace system will take control over a number of pins, irrespectively of the GPIO configuration. Three different OCD trace pin mappings are possible,

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
DP	USB Device Port Data +	Analog		
VBUS	USB VBUS Monitor and OTG Negociation	Analog Input		
ID	ID Pin of the USB Bus	Input		
VBOF	USB VBUS On/off: bus power control port	output		

3.4 I/O Line Considerations

3.4.1 JTAG pins

The JTAG is enabled if TCK is low while the RESET_N pin is released. The TCK, TMS, and TDI pins have pull-up resistors when JTAG is enabled. The TCK pin always have pull-up enabled during reset. The TDO pin is an output, driven at VDDIO1, and has no pull-up resistor. The JTAG pins can be used as GPIO pins and muxed with peripherals when the JTAG is disabled. Please refer to Section 3.2.4 for the JTAG port connections.

3.4.2 RESET_N pin

The RESET_N pin integrates a pull-up resistor to VDDIO1. As the product integrates a power-on reset cell, the RESET_N pin can be left unconnected in case no reset from the system needs to be applied to the product.

The RESET_N pin is also used for the aWire debug protocol. When the pin is used for debugging, it must not be driven by external circuitry.

3.4.3 TWI pins

When these pins are used for TWI, the pins are open-drain outputs with slew-rate limitation and inputs with inputs with spike-filtering. When used as GPIO-pins or used for other peripherals, the pins have the same characteristics as GPIO pins.

3.4.4 GPIO pins

All I/O lines integrate programmable pull-up and pull-down resistors. Most I/O lines integrate drive strength control, see Table 3-1. Programming of this pull-up and pull-down resistor or this drive strength is performed independently for each I/O line through the GPIO Controllers.

After reset, I/O lines default as inputs with pull-up/pull-down resistors disabled. After reset, output drive strength is configured to the lowest value to reduce global EMI of the device.

When the I/O line is configured as analog function (ADC I/O, AC inputs, DAC I/O), the pull-up and pull-down resistors are automatically disabled.

4. Processor and Architecture

Rev: 2.1.2.0

This chapter gives an overview of the AVR32UC CPU. AVR32UC is an implementation of the AVR32 architecture. A summary of the programming model, instruction set, and MPU is presented. For further details, see the *AVR32 Architecture Manual* and the *AVR32UC Technical Reference Manual*.

4.1 Features

- **32-bit load/store AVR32A RISC architecture**
  - **15 general-purpose 32-bit registers**
  - **32-bit Stack Pointer, Program Counter and Link Register reside in register file**
  - **Fully orthogonal instruction set**
  - **Privileged and unprivileged modes enabling efficient and secure operating systems**
  - **Innovative instruction set together with variable instruction length ensuring industry leading code density**
  - **DSP extension with saturating arithmetic, and a wide variety of multiply instructions**
- **3-stage pipeline allowing one instruction per clock cycle for most instructions**
  - **Byte, halfword, word, and double word memory access**
  - **Multiple interrupt priority levels**
- **MPU allows for operating systems with memory protection**
- **FPU enables hardware accelerated floating point calculations**
- **Secure State for supporting FlashVault technology**

4.2 AVR32 Architecture

AVR32 is a new, high-performance 32-bit RISC microprocessor architecture, designed for cost-sensitive embedded applications, with particular emphasis on low power consumption and high code density. In addition, the instruction set architecture has been tuned to allow a variety of microarchitectures, enabling the AVR32 to be implemented as low-, mid-, or high-performance processors. AVR32 extends the AVR family into the world of 32- and 64-bit applications.

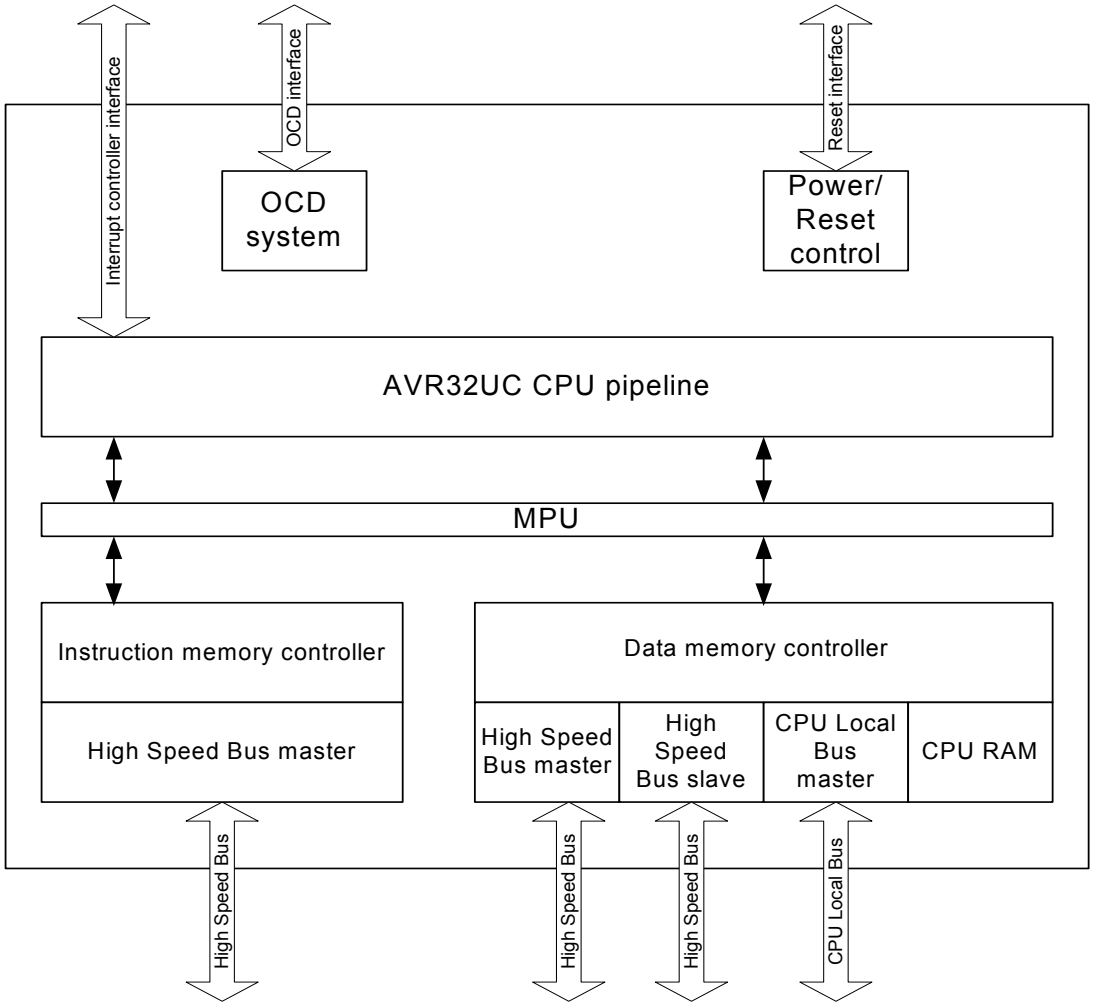
Through a quantitative approach, a large set of industry recognized benchmarks has been compiled and analyzed to achieve the best code density in its class. In addition to lowering the memory requirements, a compact code size also contributes to the core's low power characteristics. The processor supports byte and halfword data types without penalty in code size and performance.

Memory load and store operations are provided for byte, halfword, word, and double word data with automatic sign- or zero extension of halfword and byte data. The C-compiler is closely linked to the architecture and is able to exploit code optimization features, both for size and speed.

In order to reduce code size to a minimum, some instructions have multiple addressing modes. As an example, instructions with immediates often have a compact format with a smaller immediate, and an extended format with a larger immediate. In this way, the compiler is able to use the format giving the smallest code size.

Another feature of the instruction set is that frequently used instructions, like add, have a compact format with two operands as well as an extended format with three operands. The larger format increases performance, allowing an addition and a data move in the same instruction in a

Figure 4-1. Overview of the AVR32UC CPU



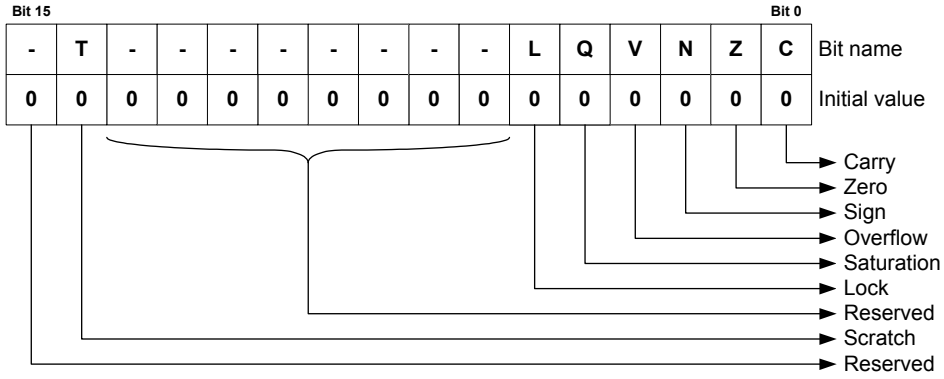
4.3.1 Pipeline Overview

AVR32UC has three pipeline stages, Instruction Fetch (IF), Instruction Decode (ID), and Instruction Execute (EX). The EX stage is split into three parallel subsections, one arithmetic/logic (ALU) section, one multiply (MUL) section, and one load/store (LS) section.

Instructions are issued and complete in order. Certain operations require several clock cycles to complete, and in this case, the instruction resides in the ID and EX stages for the required number of clock cycles. Since there is only three pipeline stages, no internal data forwarding is required, and no data dependencies can arise in the pipeline.

Figure 4-2 on page 28 shows an overview of the AVR32UC pipeline stages.

Figure 4-5. The Status Register Low Halfword



4.4.3 Processor States

4.4.3.1 Normal RISC State

The AVR32 processor supports several different execution contexts as shown in Table 4-2.

Table 4-2. Overview of Execution Modes, their Priorities and Privilege Levels.

Priority	Mode	Security	Description
1	Non Maskable Interrupt	Privileged	Non Maskable high priority interrupt mode
2	Exception	Privileged	Execute exceptions
3	Interrupt 3	Privileged	General purpose interrupt mode
4	Interrupt 2	Privileged	General purpose interrupt mode
5	Interrupt 1	Privileged	General purpose interrupt mode
6	Interrupt 0	Privileged	General purpose interrupt mode
N/A	Supervisor	Privileged	Runs supervisor calls
N/A	Application	Unprivileged	Normal program execution mode

Mode changes can be made under software control, or can be caused by external interrupts or exception processing. A mode can be interrupted by a higher priority mode, but never by one with lower priority. Nested exceptions can be supported with a minimal software overhead.

When running an operating system on the AVR32, user processes will typically execute in the application mode. The programs executed in this mode are restricted from executing certain instructions. Furthermore, most system registers together with the upper halfword of the status register cannot be accessed. Protected memory areas are also not available. All other operating modes are privileged and are collectively called System Modes. They have full access to all privileged and unprivileged resources. After a reset, the processor will be in supervisor mode.

4.4.3.2 Debug State

The AVR32 can be set in a debug state, which allows implementation of software monitor routines that can read out and alter system information for use during application development. This implies that all system and application registers, including the status registers and program counters, are accessible in debug state. The privileged instructions are also available.

All interrupt levels are by default disabled when debug state is entered, but they can individually be switched on by the monitor routine by clearing the respective mask bit in the status register.

Table 5-2. Flash Memory Parameters

Part Number	Flash Size (FLASH_PW)	Number of pages (FLASH_P)	Page size (FLASH_W)
AT32UC3C0512C AT32UC3C1512C AT32UC3C2512C	512 Kbytes	1024	128 words
AT32UC3C0256C AT32UC3C1256C AT32UC3C2256C	256 Kbytes	512	128 words
AT32UC3C0128C AT32UC3C1128C AT32UC3C2128C	128 Kbytes	256	128 words
AT32UC3C064C AT32UC3C164C AT32UC3C264C	64 Kbytes	128	128 words

5.3 Peripheral Address Map

Table 5-3. Peripheral Address Mapping

Address		Peripheral Name
0xFFFD0000	PDCA	Peripheral DMA Controller - PDCA
0xFFFD1000	MDMA	Memory DMA - MDMA
0xFFFD1400	USART1	Universal Synchronous/Asynchronous Receiver/Transmitter - USART1
0xFFFD1800	SPI0	Serial Peripheral Interface - SPI0
0xFFFD1C00	CANIF	Control Area Network interface - CANIF
0xFFFD2000	TC0	Timer/Counter - TC0
0xFFFD2400	ADCIFA	ADC controller interface with Touch Screen functionality - ADCIFA
0xFFFD2800	USART4	Universal Synchronous/Asynchronous Receiver/Transmitter - USART4
0xFFFD2C00	TWIM2	Two-wire Master Interface - TWIM2
0xFFFD3000	TWIS2	Two-wire Slave Interface - TWIS2

6. Supply and Startup Considerations

6.1 Supply Considerations

6.1.1 Power Supplies

The AT32UC3C has several types of power supply pins:

- **VDDIO pins (VDDIO1, VDDIO2, VDDIO3): Power I/O lines. Two voltage ranges are available: 5V or 3.3V nominal. The VDDIO pins should be connected together.**
- **VDDANA: Powers the Analog part of the device (Analog I/Os, ADC, ACs, DACs). 2 voltage ranges available: 5V or 3.3V nominal.**
- **VDDIN_5: Input voltage for the 1.8V and 3.3V regulators. Two Voltage ranges are available: 5V or 3.3V nominal.**
- **VDDIN_33:**
  - **USB I/O power supply**
  - **if the device is 3.3V powered: Input voltage, voltage is 3.3V nominal.**
  - **if the device is 5V powered: stabilization for the 3.3V voltage regulator, requires external capacitors**
- **VDDCORE: Stabilization for the 1.8V voltage regulator, requires external capacitors.**
- **GNDCORE: Ground pins for the voltage regulators and the core.**
- **GNDANA: Ground pin for Analog part of the design**
- **GNDPLL: Ground pin for the PLLs**
- **GNDIO pins (GNDIO1, GNDIO2, GNDIO3): Ground pins for the I/O lines. The GNDIO pins should be connected together.**

See ”Electrical Characteristics” on page 50 for power consumption on the various supply pins.

For decoupling recommendations for the different power supplies, please refer to the schematic checklist.

6.1.2 Voltage Regulators

The AT32UC3C embeds two voltage regulators:

- One 1.8V internal regulator that converts from VDDIN_5 to 1.8V. The regulator supplies the output voltage on VDDCORE.
- One 3.3V internal regulator that converts from VDDIN_5 to 3.3V. The regulator supplies the USB pads on VDDIN_33. If the USB is not used or if VDDIN_5 is within the 3V range, the 3.3V regulator can be disabled through the VREG33CTL field of the VREGCTRL SCIF register.

6.1.3 Regulators Connection

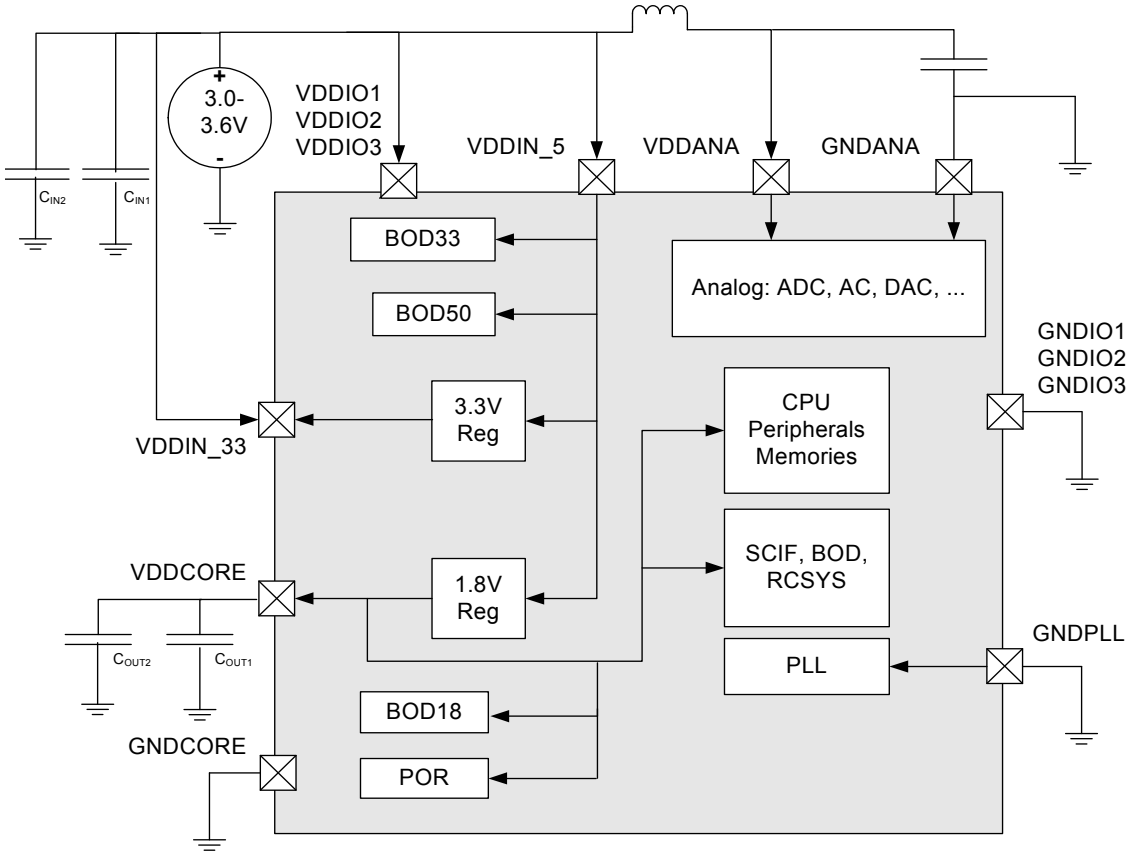
The AT32UC3C supports two power supply configurations.

- 5V single supply mode
- 3.3V single supply mode

6.1.3.1 5V Single Supply Mode

In 5V single supply mode, the 1.8V internal regulator is connected to the 5V source (VDDIN_5 pin) and its output feeds VDDCORE.

Figure 6-2. 3 Single Power Supply Mode



6.1.4 Power-up Sequence

6.1.4.1 Maximum Rise Rate

To avoid risk of latch-up, the rise rate of the power supplies must not exceed the values described in Table 7-2 on page 51 .

Recommended order for power supplies is also described in this table.

6.1.4.2 Minimum Rise Rate

The integrated Power-Reset circuitry monitoring the powering supply requires a minimum rise rate for the VDDIN_5 power supply.

See Table 7-2 on page 51 for the minimum rise rate value.

If the application can not ensure that the minimum rise rate condition for the VDDIN power supply is met, the following configuration can be used:

- A logic “0” value is applied during power-up on pin RESET_N until:
  - VDDIN_5 rises above 4.5V in 5V single supply mode.
  - VDDIN_33 rises above 3V in 3.3V single supply mode.

  - Internal 3.3V regulator is off
- T_A = 25°C
- I/Os are configured as inputs, with internal pull-up enabled.
- Oscillators
  - OSC0/1 (crystal oscillator) stopped
  - OSC32K (32KHz crystal oscillator) stopped
  - PLL0 running
  - PLL1 stopped
- Clocks
  - External clock on XIN0 as main clock source **(**10MHz**)**
  - CPU, HSB, and PBB clocks undivided
  - PBA, PBC clock divided by 4
  - All peripheral clocks running

Table 7-4. Power Consumption for Different Operating Modes

Mode	Conditions	Measured on	Consumption Typ	Unit
Active[1]	CPU running a recursive Fibonacci algorithm	Amp	512	µA/MHz
Idle[1]			258	
Frozen[1]			106	
Standby[1]			48	
Stop			73	µA
DeepStop			43	
Static	OSC32K and AST running		32	
	AST and OSC32K stopped		31	

Note: 1. These numbers are valid for the measured condition only and must not be extrapolated to other frequencies.

7.6.3 Phase Lock Loop (PLL0 and PLL1) Characteristics

Table 7-11. PLL Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{VCO}	Output frequency		80		240	MHz
f_{IN}	Input frequency		4		16	MHz
I_{PLL}	Current consumption	Active mode, f_{VCO} = 80MHz		250		µA
		Active mode, f_{VCO} = 240MHz		600		
$t_{STARTUP}$	Startup time, from enabling the PLL until the PLL is locked	Wide Bandwidth mode disabled		15		µs
		Wide Bandwidth mode enabled		45		

7.6.4 120MHz RC Oscillator (RC120M) Characteristics

Table 7-12. Internal 120MHz RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Output frequency[1]		88	120	152	MHz
I_{RC120M}	Current consumption			1.85		mA
$t_{STARTUP}$	Startup time			3		µs

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

7.6.5 System RC Oscillator (RCSYS) Characteristics

Table 7-13. System RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Output frequency	Calibrated at T_A = 85°C	110	115.2	120	kHz
		T_A = 25°C	105	109	115	
		T_A = -40°C	100	104	108	

7.6.6 8MHz/1MHz RC Oscillator (RC8M) Characteristics

Table 7-14. 8MHz/1MHz RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Output frequency	SCIF.RCCR8.FREQMODE = 0[1]	7.6	8	8.4	MHz
		SCIF.RCCR8.FREQMODE = 1[1]	0.955	1	1.045	
$t_{STARTUP}$	Startup time				20	µs

Notes: 1. Please refer to the SCIF chapter for details.

Table 7-29. ADC Decoupling requirements

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$C_{ADCREFPN}$	ADCREFP/ADCREFN capacitance	No voltage reference appplied on ADCREFP/ADCREFN		100		nF

Table 7-30. ADC Inputs

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{ADCINn}	ADC input voltage range		0		V_{VDDANA}	V
C_{ONCHIP}	Internal Capacitance	ADC used without S/H			5	pF
		ADC used with S/H			4	
R_{ONCHIP}	Switch resistance	ADC used without S/H			5.1	kΩ
		ADC used with S/H			4.6	

Figure 7-3. ADC input

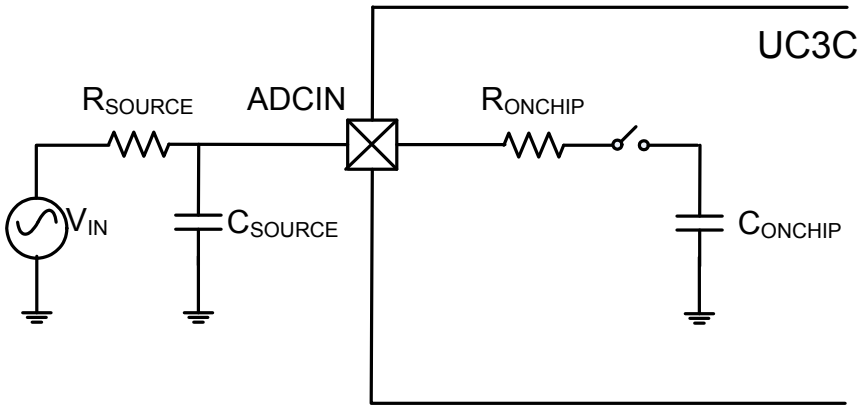


Table 7-31. ADC Transfer Characteristics 12-bit Resolution Mode[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RES	Resolution	Differential mode, V_{VDDANA} = 3V, $V_{ADCREF0}$ = 1V, ADCFIA.SEQCFGn.SRES = 0 (F_{adc} = 1.2MHz)			12	Bit
INL	Integral Non-Linearity				5	LSB
DNL	Differential Non-Linearity				3	LSB
	Offset error		-7		7	mV
	Gain error		-20		20	mV

Table 7-31. ADC Transfer Characteristics (Continued)12-bit Resolution Mode[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RES	Resolution	Differential mode, V_{VDDANA} = 5V, $V_{ADCREF0}$ = 3V, ADCFIA.SEQCFGn.SRES = 0 (F_{adc} = 1.5MHz)			12	Bit
INL	Integral Non-Linearity				4	LSB
DNL	Differential Non-Linearity				3	LSB
	Offset error		-15		15	mV
	Gain error		-25		25	mV

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-32. ADC Transfer Characteristics 10-bit Resolution Mode[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RES	Resolution	Differential mode, V_{VDDANA} = 3V, $V_{ADCREF0}$ = 1V, ADCFIA.SEQCFGn.SRES = 1 (F_{adc} = 1.5MHz)			10	Bit
INL	Integral Non-Linearity				1.25	LSB
DNL	Differential Non-Linearity				1	LSB
	Offset error		-10		10	mV
	Gain error		-20		20	mV
RES	Resolution	Differential mode, V_{VDDANA} = 5V, $V_{ADCREF0}$= 3V, ADCFIA.SEQCFGn.SRES = 1 (F_{adc} = 1.5MHz)			10	Bit
INL	Integral Non-Linearity				1.25	LSB
DNL	Differential Non-Linearity				1	LSB
	Offset error		-15		15	mV
	Gain error		-20		20	mV

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-33. ADC Transfer Characteristics 8-bit Resolution Mode[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RES	Resolution	Differential mode, V_{VDDANA} = 3V, $V_{ADCREF0}$ = 1V, ADCFIA.SEQCFGn.SRES = 2 (F_{adc} =1.5MHz)			8	Bit
INL	Integral Non-Linearity				0.3	LSB
DNL	Differential Non-Linearity				0.25	LSB
	Offset error		-10		10	mV
	Gain error		-20		20	mV
RES	Resolution	Differential mode, V_{VDDANA} = 5V, $V_{ADCREF0}$ = 3V, ADCFIA.SEQCFGn.SRES = 2 (F_{adc} = 1.5MHz)			8	Bit
INL	Integral Non-Linearity				0.2	LSB
DNL	Differential Non-Linearity				0.2	LSB
	Offset error		-20		20	mV
	Gain error		-20		20	mV

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-36. ADC and S/H Transfer Characteristics (Continued)10-bit Resolution Mode and S/H gain from 1 to 16[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RES	Resolution	Differential mode, V_{VDDANA} = 5V, $V_{ADCREF0}$ = 3V, ADCFIA.SEQCFGn.SRES = 1, S/H gain from 1 to 16 (F_{adc} = 1.5MHz)			10	Bit
INL	Integral Non-Linearity				1.5	LSB
DNL	Differential Non-Linearity				1.5	LSB
	Offset error		-25		25	mV
	Gain error		-15		15	mV

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

7.8.7 Digital to Analog Converter (DAC) Characteristics

Table 7-37. Channel Conversion Time and DAC Clock

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{DAC}	DAC clock frequency				1	MHz
$t_{STARTUP}$	Startup time				3	µs
t_{CONV}	Conversion time (latency)	No S/H enabled, internal DAC			1	µs
		One S/H			1.5	µs
		Two S/H			2	µs
	Throughput rate				$1/t_{CONV}$	MSPS

Table 7-38. External Voltage Reference Input

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{DACREF}	DACREF input voltage range		1.2		V_{VDDANA}-0.7	V

Table 7-39. DAC Outputs

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	Output range	with external DAC reference	0.2		V_{DACREF}	V
		with internal DAC reference	0.2		V_{VDDANA}-0.7	
C_{LOAD}	Output capacitance		0		100	pF
R_{LOAD}	Output resitance		2			kΩ

Figure 7-4. DAC output

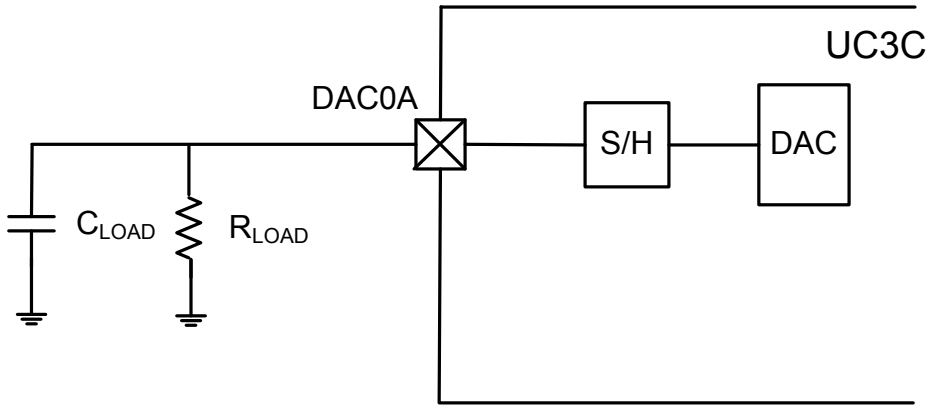


Table 7-40. Transfer Characteristics[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Units
RES	Resolution	V_{VDDANA} = 3V, V_{DACREF} = 2V, One S/H			12	Bit
INL	Integral Non-Linearity			8		LSB
DNL	Differential Non-linearity			6		LSB
	Offset error		-30		30	mV
	Gain error		-30		30	mV
RES	Resolution	V_{VDDANA} = 5V, V_{DACREF} = 3V, One S/H			12	Bit
INL	Integral Non-Linearity			12		LSB
DNL	Differential Non-linearity			6		LSB
	Offset error		-30		30	mV
	Gain error		-30		30	mV

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

7.9.4.2 Slave mode

Figure 7-13. SPI Slave Mode With (CPOL= 0 and NCPHA= 1) or (CPOL= 1 and NCPHA= 0)

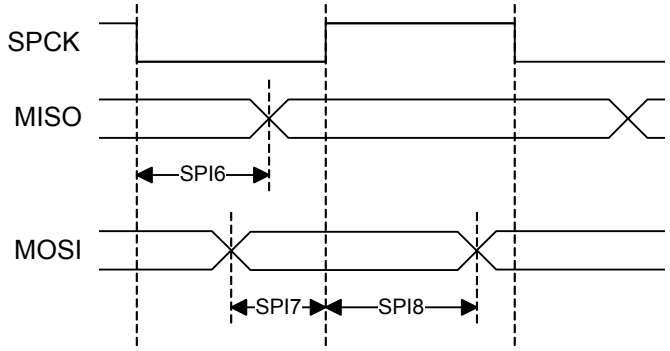


Figure 7-14. SPI Slave Mode With (CPOL= NCPHA= 0) or (CPOL= NCPHA= 1)

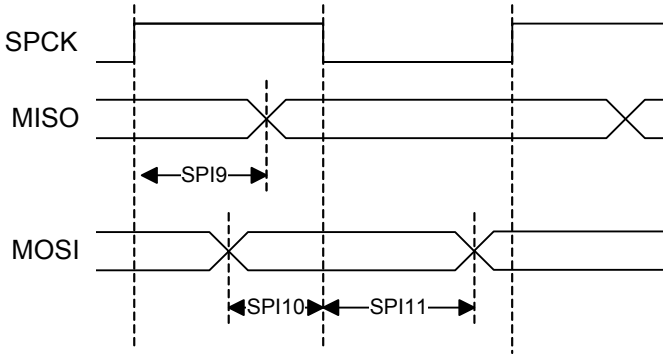


Figure 7-15. SPI Slave Mode NPCS Timing

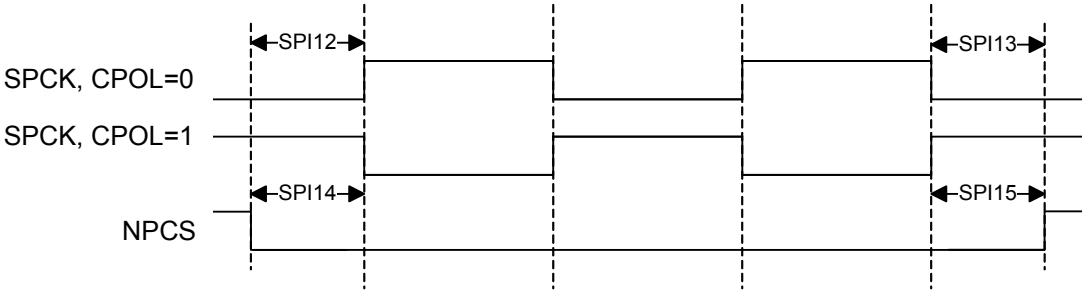


Table 7-49. SPI Timing, Slave Mode[1]

Symbol	Parameter	Conditions	Min	Max	Units
SPI6	SPCK falling to MISO delay	external capacitor = 40pF		29	ns
SPI7	MOSI setup time before SPCK rises		0		ns
SPI8	MOSI hold time after SPCK rises		6.5		ns
SPI9	SPCK rising to MISO delay			30	ns
SPI10	MOSI setup time before SPCK falls		0		ns
SPI11	MOSI hold time after SPCK falls		5		ns
SPI12	NPCS setup time before SPCK rises		0		ns
SPI13	NPCS hold time after SPCK falls		1.5		ns
SPI14	NPCS setup time before SPCK falls		0		ns
SPI15	NPCS hold time after SPCK rises		1.5		ns

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

Maximum SPI Frequency, Slave Input Mode

The maximum SPI slave input frequency is given by the following formula:

$$f_{SPCKMAX} = MIN(f_{CLKSPI}, \frac{1}{SPIn})$$

Where $SPIn$ is the MOSI setup and hold time, SPI7 + SPI8 or SPI10 + SPI11 depending on CPOL and NCPHA. f_{CLKSPI} is the maximum frequency of the CLK_SPI. Refer to the SPI chapter for a description of this clock.

Maximum SPI Frequency, Slave Output Mode

The maximum SPI slave output frequency is given by the following formula:

$$f_{SPCKMAX} = MIN(f_{PINMAX}, \frac{1}{SPIn + t_{SETUP}})$$

Where $SPIn$ is the MISO delay, SPI6 or SPI9 depending on CPOL and NCPHA. t_{SETUP} is the SPI master setup time. Please refer to the SPI masterdatasheet for t_{SETUP} .f_{PINMAX} is the maximum frequency of the SPI pins. Please refer to the I/O Pin Characteristics section for the maximum frequency of the pins.

7.9.5 TWIM/TWIS Timing

Figure 7-50 shows the TWI-bus timing requirements and the compliance of the device with them. Some of these requirements (t_r and t_f) are met by the device without requiring user intervention. Compliance with the other requirements (t_{HD-STA}, t_{SU-STA}, t_{SU-STO}, t_{HD-DAT}, $t_{SU-DAT-I2C}$, $t_{LOW-I2C}$, t_{HIGH}, and f_{TWCK}) requires user intervention through appropriate programming of the relevant

10.2.12 WDT

1 **Clearing the Watchdog Timer (WDT) counter in second half of timeout period will issue a Watchdog reset**

If the WDT counter is cleared in the second half of the timeout period, the WDT will immediately issue a Watchdog reset.

Fix/Workaround

Use twice as long timeout period as needed and clear the WDT counter within the first half of the timeout period. If the WDT counter is cleared after the first half of the timeout period, you will get a Watchdog reset immediately. If the WDT counter is not cleared at all, the time before the reset will be twice as long as needed.

2 **WDT Control Register does not have synchronization feedback**

When writing to the Timeout Prescale Select (PSEL), Time Ban Prescale Select (TBAN), Enable (EN), or WDT Mode (MODE) fieldss of the WDT Control Register (CTRL), a synchronizer is started to propagate the values to the WDT clcok domain. This synchronization takes a finite amount of time, but only the status of the synchronization of the EN bit is reflected back to the user. Writing to the synchronized fields during synchronization can lead to undefined behavior.

Fix/Workaround

-When writing to the affected fields, the user must ensure a wait corresponding to 2 clock cycles of both the WDT peripheral bus clock and the selected WDT clock source.

-When doing writes that changes the EN bit, the EN bit can be read back until it reflects the written value.