

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

### Details

Product Status	Obsolete
Core Processor	C166SV2
Core Size	16/32-Bit
Speed	128MHz
Connectivity	CANbus, EBI/EMI, I <sup>2</sup> C, LINbus, SPI, SSC, UART/USART, USI
Peripherals	I <sup>2</sup> S, POR, PWM, WDT
Number of I/O	75
Program Memory Size	832KB (832K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	98K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP Exposed Pad
Supplier Device Package	PG-LQFP-100-8
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/xc2361e104f128laakfuma1

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



- -

VAAAA

Revisio	n History: V1.3 2014-06
Previous	s Versions:
V1.2 20	12-06
V1.1 20 <sup>4</sup>	11-10
V1.0 20	10-12
Page	Subjects (major changes since last revision)
9	Added XC2361E-136FxxLR to Basic Device Types
10	Moved XC2367E-136FxxL from Basic to Special Device Types. Added XC2367E-136FxxLR to Special Device Types
127	Added package type PG-LQFP-100-15

## Trademarks

C166<sup>™</sup>, TriCore<sup>™</sup> and DAVE<sup>™</sup> are trademarks of Infineon Technologies AG.

### We Listen to Your Comments

Is there any information in this document that you feel is wrong, unclear or missing? Your feedback will help us to continuously improve the quality of this document. Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com





### **General Device Information**

# 2.1 Pin Configuration and Definition

The pins of the XC236xE are described in detail in **Table 6**, which includes all alternate functions. For further explanations please refer to the footnotes at the end of the table. The following figure summarizes all pins, showing their locations on the four sides of the package.



Figure 2 XC236xE Pin Configuration (top view)



### **General Device Information**

Table	able 6Pin Definitions and Functions (cont'd)							
Pin	Symbol	Ctrl.	Туре	Function				
69	P10.4	O0 / I	St/B	Bit 4 of Port 10, General Purpose Input/Output				
	U0C0_SELO 3	O1	St/B	USIC0 Channel 0 Select/Control 3 Output				
	CCU60_COU T61	O2	St/B	CCU60 Channel 1 Output				
	AD4	OH / IH	St/B	External Bus Interface Address/Data Line 4				
	U0C0_DX2B	I	St/B	USIC0 Channel 0 Shift Control Input				
	U0C1_DX2B	I	St/B	USIC0 Channel 1 Shift Control Input				
	ESR1_9	I	St/B	ESR1 Trigger Input 9				
70	P10.5	O0 / I	St/B	Bit 5 of Port 10, General Purpose Input/Output				
	U0C1_SCLK OUT	O1	St/B	USIC0 Channel 1 Shift Clock Output				
	CCU60_COU T62	O2	St/B	CCU60 Channel 2 Output				
	U2C0_DOUT	O3	St/B	USIC2 Channel 0 Shift Data Output				
	AD5	OH / IH	St/B	External Bus Interface Address/Data Line 5				
	U0C1_DX1B	I	St/B	USIC0 Channel 1 Shift Clock Input				
71	P0.6	O0 / I	St/B	Bit 6 of Port 0, General Purpose Input/Output				
	U1C1_DOUT	01	St/B	USIC1 Channel 1 Shift Data Output				
	TxDC1	O2	St/B	CAN Node 1 Transmit Data Output				
	CCU61_COU T63	O3	St/B	CCU61 Channel 3 Output				
	A6	ОН	St/B	External Bus Interface Address Line 6				
	U1C1_DX0A	I	St/B	USIC1 Channel 1 Shift Data Input				
	CCU61_CTR APA	1	St/B	CCU61 Emergency Trap Input				
	U1C1_DX1B	I	St/B	USIC1 Channel 1 Shift Clock Input				



With this hardware most XC236xE instructions are executed in a single machine cycle of ns @ -MHz CPU clock. For example, shift and rotate instructions are always processed during one machine cycle, no matter how many bits are shifted. Also, multiplication and most MAC instructions execute in one cycle. All multiple-cycle instructions have been optimized so that they can be executed very fast; for example, a 32-/16-bit division is started within 4 cycles while the remaining cycles are executed in the background. Another pipeline optimization, the branch target prediction, eliminates the execution time of branch instructions if the prediction was correct.

The CPU has a register context consisting of up to three register banks with 16 wordwide GPRs each at its disposal. One of these register banks is physically allocated within the on-chip DPRAM area. A Context Pointer (CP) register determines the base address of the active register bank accessed by the CPU at any time. The number of these register bank copies is only restricted by the available internal RAM space. For easy parameter passing, a register bank may overlap others.

A system stack of up to 32 Kwords is provided for storage of temporary data. The system stack can be allocated to any location within the address space (preferably in the on-chip RAM area); it is accessed by the CPU with the stack pointer (SP) register. Two separate SFRs, STKOV and STKUN, are implicitly compared with the stack pointer value during each stack access to detect stack overflow or underflow.

The high performance of the CPU hardware implementation can be best utilized by the programmer with the highly efficient XC236xE instruction set. This includes the following instruction classes:

- Standard Arithmetic Instructions
- DSP-Oriented Arithmetic Instructions
- Logical Instructions
- Boolean Bit Manipulation Instructions
- Compare and Loop Control Instructions
- Shift and Rotate Instructions
- Prioritize Instruction
- Data Movement Instructions
- System Stack Instructions
- Jump and Call Instructions
- Return Instructions
- System Control Instructions
- Miscellaneous Instructions

The basic instruction length is either 2 or 4 bytes. Possible operand types are bits, bytes and words. A variety of direct, indirect or immediate addressing modes are provided to specify the required operands.



# 3.4 Memory Protection Unit (MPU)

The XC236xE's Memory Protection Unit (MPU) protects user-specified memory areas from unauthorized read, write, or instruction fetch accesses. The MPU can protect the whole address space including the peripheral area. This completes established mechanisms such as the register security mechanism or stack overrun/underrun detection.

Four Protection Levels support flexible system programming where operating system, low level drivers, and applications run on separate levels. Each protection level permits different access restrictions for instructions and/or data.

Every access is checked (if the MPU is enabled) and an access violating the permission rules will be marked as invalid and leads to a protection trap.

A set of protection registers for each protection level specifies the address ranges and the access permissions. Applications requiring more than 4 protection levels can dynamically re-program the protection registers.

## 3.5 Memory Checker Module (MCHK)

The XC236xE's Memory Checker Module calculates a checksum (fractional polynomial division) on a block of data, often called Cyclic Redundancy Code (CRC). It is based on a 32-bit linear feedback shift register and may, therefore, also be used to generate pseudo-random numbers.

The Memory Checker Module is a 16-bit parallel input signature compression circuitry which enables error detection within a block of data stored in memory, registers, or communicated e.g. via serial communication lines. It reduces the probability of error masking due to repeated error patterns by calculating the signature of blocks of data.

The polynomial used for operation is configurable, so most of the commonly used polynomials may be used. Also, the block size for generating a CRC result is configurable via a local counter. An interrupt may be generated if testing the current data block reveals an error.

An autonomous CRC compare circuitry is included to enable redundant error detection, e.g. to enable higher safety integrity levels.

The Memory Checker Module provides enhanced fault detection (beyond parity or ECC) for data and instructions in volatile and non volatile memories. This is especially important for the safety and reliability of embedded systems.

to a dedicated vector table location). The occurrence of a hardware trap is also indicated by a single bit in the trap flag register (TFR). Unless another higher-priority trap service is in progress, a hardware trap will interrupt any ongoing program execution. In turn, hardware trap services can normally not be interrupted by standard or PEC interrupts.

Depending on the package option up to 3 External Service Request (ESR) pins are provided. The ESR unit processes their input values and allows to implement user controlled trap functions (System Requests SR0 and SR1). In this way reset, wakeup and power control can be efficiently realized.

Software interrupts are supported by the 'TRAP' instruction in combination with an individual trap (interrupt) number. Alternatively to emulate an interrupt by software a program can trigger interrupt requests by writing the Interrupt Request (IR) bit of an interrupt control register.

# 3.7 On-Chip Debug Support (OCDS)

The On-Chip Debug Support system built into the XC236xE provides a broad range of debug and emulation features. User software running on the XC236xE can be debugged within the target system environment.

The OCDS is controlled by an external debugging device via the debug interface. This either consists of the 2-pin Device Access Port (DAP) or of the JTAG port conforming to IEEE-1149. The debug interface can be completed with an optional break interface.

The debugger controls the OCDS with a set of dedicated registers accessible via the debug interface (DAP or JTAG). In addition the OCDS system can be controlled by the CPU, e.g. by a monitor program. An injection interface allows the execution of OCDS-generated instructions by the CPU.

Multiple breakpoints can be triggered by on-chip hardware, by software, or by an external trigger input. Single stepping is supported, as is the injection of arbitrary instructions and read/write access to the complete internal address space. A breakpoint trigger can be answered with a CPU halt, a monitor call, a data transfer, or/and the activation of an external signal.

Tracing of data can be obtained via the debug interface, or via the external bus interface for increased performance.

Tracing of program execution is supported by the XC2000 Family emulation device. With this device the DAP can operate on clock rates of up to 20 MHz.

The DAP interface uses two interface signals, the JTAG interface uses four interface signals, to communicate with external circuitry. The debug interface can be amended with two optional break lines.



# 3.10 General Purpose Timer (GPT12E) Unit

The GPT12E unit is a very flexible multifunctional timer/counter structure which can be used for many different timing tasks such as event timing and counting, pulse width and duty cycle measurements, pulse generation, or pulse multiplication.

The GPT12E unit incorporates five 16-bit timers organized in two separate modules, GPT1 and GPT2. Each timer in each module may either operate independently in a number of different modes or be concatenated with another timer of the same module.

Each of the three timers T2, T3, T4 of **module GPT1** can be configured individually for one of four basic modes of operation: Timer, Gated Timer, Counter, and Incremental Interface Mode. In Timer Mode, the input clock for a timer is derived from the system clock and divided by a programmable prescaler. Counter Mode allows timer clocking in reference to external events.

Pulse width or duty cycle measurement is supported in Gated Timer Mode, where the operation of a timer is controlled by the 'gate' level on an external input pin. For these purposes each timer has one associated port pin (TxIN) which serves as a gate or clock input. The maximum resolution of the timers in module GPT1 is 4 system clock cycles.

The counting direction (up/down) for each timer can be programmed by software or altered dynamically by an external signal on a port pin (TxEUD), e.g. to facilitate position tracking.

In Incremental Interface Mode the GPT1 timers can be directly connected to the incremental position sensor signals A and B through their respective inputs TxIN and TxEUD. Direction and counting signals are internally derived from these two input signals, so that the contents of the respective timer Tx corresponds to the sensor position. The third position sensor signal TOP0 can be connected to an interrupt input.

Timer T3 has an output toggle latch (T3OTL) which changes its state on each timer overflow/underflow. The state of this latch may be output on pin T3OUT e.g. for time out monitoring of external hardware components. It may also be used internally to clock timers T2 and T4 for measuring long time periods with high resolution.

In addition to the basic operating modes, T2 and T4 may be configured as reload or capture register for timer T3. A timer used as capture or reload register is stopped. The contents of timer T3 is captured into T2 or T4 in response to a signal at the associated input pin (TxIN). Timer T3 is reloaded with the contents of T2 or T4, triggered either by an external signal or a selectable state transition of its toggle latch T3OTL. When both T2 and T4 are configured to alternately reload T3 on opposite state transitions of T3OTL with the low and high times of a PWM signal, this signal can be continuously generated without software intervention.



# 3.11 Real Time Clock

The Real Time Clock (RTC) module of the XC236xE can be clocked with a clock signal selected from internal sources or external sources (pins).

The RTC basically consists of a chain of divider blocks:

- Selectable 32:1 and 8:1 dividers (on off)
- The reloadable 16-bit timer T14
- The 32-bit RTC timer block (accessible via registers RTCH and RTCL) consisting of: – a reloadable 10-bit timer
  - a reloadable 6-bit timer
  - a reloadable 6-bit timer
  - a reloadable 10-bit timer

All timers count up. Each timer can generate an interrupt request. All requests are combined to a common node request.



## Figure 7 RTC Block Diagram

Note: The registers associated with the RTC are only affected by a power reset.



The RTC module can be used for different purposes:

- System clock to determine the current time and date
- Cyclic time-based interrupt, to provide a system time tick independent of CPU frequency and other resources
- 48-bit timer for long-term measurements
- Alarm interrupt at a defined time



## 3.12 A/D Converters

For analog signal measurement, up to two 10-bit A/D converters (ADC0, ADC1) with 11 + 5 multiplexed input channels and a sample and hold circuit have been integrated on-chip. 4 inputs can be converted by both A/D converters. Conversions use the successive approximation method. The sample time (to charge the capacitors) and the conversion time are programmable so that they can be adjusted to the external circuit. The A/D converters can also operate in 8-bit conversion mode, further reducing the conversion time.

Several independent conversion result registers, selectable interrupt requests, and highly flexible conversion sequences provide a high degree of programmability to meet the application requirements. Both modules can be synchronized to allow parallel sampling of two input channels.

For applications that require more analog input channels, external analog multiplexers can be controlled automatically. For applications that require fewer analog input channels, the remaining channel inputs can be used as digital input port pins.

The A/D converters of the XC236xE support two types of request sources which can be triggered by several internal and external events.

- Parallel requests are activated at the same time and then executed in a predefined sequence.
- Queued requests are executed in a user-defined sequence.

In addition, the conversion of a specific channel can be inserted into a running sequence without disturbing that sequence. All requests are arbitrated according to the priority level assigned to them.

Data reduction features reduce the number of required CPU access operations allowing the precise evaluation of analog inputs (high conversion rate) even at a low CPU speed. Result data can be reduced by limit checking or accumulation of results.

The Peripheral Event Controller (PEC) can be used to control the A/D converters or to automatically store conversion results to a table in memory for later evaluation, without requiring the overhead of entering and exiting interrupt routines for each data transfer. Each A/D converter contains eight result registers which can be concatenated to build a result FIFO. Wait-for-read mode can be enabled for each result register to prevent the loss of conversion data.

In order to decouple analog inputs from digital noise and to avoid input trigger noise, those pins used for analog input can be disconnected from the digital input stages. This can be selected for each pin separately with the Port x Digital Input Disable registers.

The Auto-Power-Down feature of the A/D converters minimizes the power consumption when no conversion is in progress.

Broken wire detection for each channel and a multiplexer test mode provide information to verify the proper operation of the analog signal sources (e.g. a sensor system).



## 3.14 MultiCAN Module

The MultiCAN module contains independently operating CAN nodes with Full-CAN functionality which are able to exchange Data and Remote Frames using a gateway function. Transmission and reception of CAN frames is handled in accordance with CAN specification V2.0 B (active). Each CAN node can receive and transmit standard frames with 11-bit identifiers as well as extended frames with 29-bit identifiers.

All CAN nodes share a common set of message objects. Each message object can be individually allocated to one of the CAN nodes. Besides serving as a storage container for incoming and outgoing frames, message objects can be combined to build gateways between the CAN nodes or to set up a FIFO buffer.

Note: The number of CAN nodes and message objects depends on the selected device type.

The message objects are organized in double-chained linked lists, where each CAN node has its own list of message objects. A CAN node stores frames only into message objects that are allocated to its own message object list and it transmits only messages belonging to this message object list. A powerful, command-driven list controller performs all message object list operations.



Figure 9 Block Diagram of MultiCAN Module



	denon Set Summary (cont d)	
Mnemonic	Description	Bytes
ROL/ROR	Rotate left/right direct word GPR	2
ASHR	Arithmetic (sign bit) shift right direct word GPR	2
MOV(B)	Move word (byte) data	2/4
MOVBS/Z	Move byte operand to word op. with sign/zero extension	2/4
JMPA/I/R	Jump absolute/indirect/relative if condition is met	4
JMPS	Jump absolute to a code segment	4
JB(C)	Jump relative if direct bit is set (and clear bit)	4
JNB(S)	Jump relative if direct bit is not set (and set bit)	4
CALLA/I/R	Call absolute/indirect/relative subroutine if condition is met	4
CALLS	Call absolute subroutine in any code segment	4
PCALL	Push direct word register onto system stack and call absolute subroutine	4
TRAP	Call interrupt service routine via immediate trap number	2
PUSH/POP	Push/pop direct word register onto/from system stack	2
SCXT	Push direct word register onto system stack and update register with word operand	4
RET(P)	Return from intra-segment subroutine (and pop direct word register from system stack)	2
RETS	Return from inter-segment subroutine	2
RETI	Return from interrupt service subroutine	2
SBRK	Software Break	2
SRST	Software Reset	4
IDLE	Enter Idle Mode	4
PWRDN	Unused instruction <sup>1)</sup>	4
SRVWDT	Service Watchdog Timer	4
DISWDT/ENWDT	Disable/Enable Watchdog Timer	4
EINIT	End-of-Initialization Register Lock	4
ATOMIC	Begin ATOMIC sequence	2
EXTR	Begin EXTended Register sequence	2
EXTP(R)	Begin EXTended Page (and Register) sequence	2/4
EXTS(R)	Begin EXTended Segment (and Register) sequence	2/4

### Table 11 Instruction Set Summary (cont'd)



## 4.2.3 Power Consumption

The power consumed by the XC236xE depends on several factors such as supply voltage, operating frequency, active circuits, and operating temperature. The power consumption specified here consists of two components:

- The switching current  $I_{\rm S}$  depends on the device activity
- The leakage current I<sub>LK</sub> depends on the device temperature

To determine the actual power consumption, always both components, switching current  $I_{\rm S}$  and leakage current  $I_{\rm LK}$  must be added:

 $I_{\text{DDP}} = I_{\text{S}} + I_{\text{LK}}.$ 

Note: The power consumption values are not subject to production test. They are verified by design/characterization.

To determine the total power consumption for dimensioning the external power supply, also the pad driver currents must be considered.

The given power consumption parameters and their values refer to specific operating conditions:

Active mode:

Regular operation, i.e. peripherals are active, code execution out of Flash.

Stopover mode:

Crystal oscillator and PLL stopped, Flash switched off, clock in domain DMP\_1 stopped.

Note: The maximum values cover the complete specified operating range of all manufactured devices.

The typical values refer to average devices under typical conditions, such as nominal supply voltage, room temperature, application-oriented activity.

After a power reset, the decoupling capacitors for  $V_{\rm DDIM}$  and  $V_{\rm DDI1}$  are charged with the maximum possible current.

For additional information, please refer to Section 5.2, Thermal Considerations.

Note: Operating Conditions apply.

Parameter	Symbol	Values			Unit	Note /
		Min.	Тур.	Max.		Test Condition
Power supply current (active) with all peripherals active and EVVRs on	I <sub>SACT</sub> CC	-	10 + 1.0 x $f_{SYS}^{1)}$	$15 + 1.5 x f_{SYS}^{1)}$	mA	2)3)
Power supply current in stopover mode, EVVRs on	I <sub>SSO</sub> CC	-	1.6	4	mA	

## Table 18 Switching Power Consumption

1)  $f_{\rm SYS}$  in MHz.





Figure 11 Supply Current in Active Mode as a Function of Frequency

Note: Operating Conditions apply.



## Table 19 Leakage Power Consumption

Parameter	Symbol	Values			Unit	Note /
		Min.	Тур.	Max.		Test Condition
Leakage supply current	I <sub>LK1</sub> CC	-	0.03	0.05	mA	$T_{\rm J}$ = 25 °C <sup>2)</sup>
(DMP_1 powered) <sup>1)</sup>		-	0.7	1.7	mA	$T_{\rm J} = 85 \ ^{\circ}{\rm C}^{2)}$
		-	3.0	8.3	mA	$T_{\rm J}$ = 125 °C <sup>2)</sup>
		-	6.4	18.5	mA	$T_{\rm J}$ = 150 °C <sup>2)</sup>

Note: A fraction of the leakage current flows through domain DMP\_A (pin  $V_{DDPA}$ ). This current can be calculated as 7 000 × e<sup>- $\alpha$ </sup>, with  $\alpha$  = 5 000 / (273 + 1.3×T<sub>J</sub>). For  $T_J$  = 150°C, this results in a current of 160  $\mu$ A.

The leakage power consumption can be calculated according to the following formula:  $I_{LK1} = 580\ 000 + e^{-\alpha}$ , with  $\alpha = 5\ 000\ /\ (273 + B \times T_J)$ 

Parameter B must be replaced by

- 1.1 for typical values
- 1.4 for maximum values



### **PLL Frequency Band Selection**

Different frequency bands can be selected for the VCO so that the operation of the PLL can be adjusted to a wide range of input and output frequencies:

Parameter	Symbol		Values		Unit	Note / Test Condition
		Min.	Тур.	Max.		
VCO output frequency	$f_{\rm VCO}{\rm CC}$	50	-	110	MHz	$VCOSEL = 00_B$
(VCO controlled)		100	-	200	MHz	VCOSEL = 01 <sub>B</sub>
		200	-	280	MHz	$VCOSEL = 10_B$
VCO output frequency	$f_{\rm VCO}$ CC	10	-	40	MHz	$VCOSEL = 00_B$
(VCO free-running)		20	-	80	MHz	$VCOSEL = 01_B$
		40	-	160	MHz	$VCOSEL = 10_B$



# 4.6.2.2 Wakeup Clock

When wakeup operation is selected (SYSCON0.CLKSEL =  $00_B$ ), the system clock is derived from the low-frequency wakeup clock source:

 $f_{SYS} = f_{WU}$ .

In this mode, a basic functionality can be maintained without requiring an external clock source and while minimizing the power consumption.

# 4.6.2.3 Selecting and Changing the Operating Frequency

When selecting a clock source and the clock generation method, the required parameters must be carefully written to the respective bitfields, to avoid unintended intermediate states.

Many applications change the frequency of the system clock ( $f_{SYS}$ ) during operation in order to optimize system performance and power consumption. Changing the operating frequency also changes the switching currents, which influences the power supply.

To ensure proper operation of the on-chip EVRs while they generate the core voltage, the operating frequency shall only be changed in certain steps. This prevents overshoots and undershoots of the supply voltage.

To avoid the indicated problems, recommended sequences are provided which ensure the intended operation of the clock system interacting with the power system. Please refer to the Programmer's Guide.



## 4.6.5 External Bus Timing

The following parameters specify the behavior of the XC236xE bus interface.

Note: These parameters are not subject to production test but verified by design and/or characterization.

Note: Operating Conditions apply.

### **Bus Interface Performance Limits**

The output frequency at the bus interface pins is limited by the performance of the output drivers. The fast clock driver (used for CLKOUT) can drive 80-MHz signals, the standard drivers can drive 40-MHz signals

Therefore, the speed of the EBC must be limited, either by limiting the system frequency to  $f_{SYS} \le 80$  MHz or by adding waitstates so that signal transitions have a minimum distance of 12.5 ns.

For a description of the bus protocol and the programming of its variable timing parameters, please refer to the User's Manual.

Parameter	Symbol		Values		Unit	Note / Test Condition
		Min.	Тур.	Max.		
CLKOUT Cycle Time <sup>1)</sup>	t <sub>5</sub> CC	-	$1/f_{\rm SYS}$	-	ns	
CLKOUT high time	t <sub>6</sub> CC	2	-	-		
CLKOUT low time	t <sub>7</sub> CC	2	-	-		
CLKOUT rise time	t <sub>8</sub> CC	-	-	3	ns	
CLKOUT fall time	t <sub>9</sub> CC	-	-	3		

### Table 32 EBC Parameters

 The CLKOUT cycle time is influenced by PLL jitter. For longer periods the relative deviation decreases (see PLL deviation formula).







Denemeter	Cumhal		Values	•	11	Note /
Parameter	Symbol	Values			Unit	Note /
		Min.	Тур.	Max.		Test Condition
TCK clock period	t <sub>1</sub> SR	50 <sup>1)</sup>	-	-	ns	
TCK high time	$t_2  \mathrm{SR}$	16	-	-	ns	
TCK low time	t <sub>3</sub> SR	16	-	-	ns	
TCK clock rise time	$t_4$ SR	-	-	8	ns	
TCK clock fall time	t <sub>5</sub> SR	-	-	8	ns	
TDI/TMS setup to TCK rising edge	t <sub>6</sub> SR	6	-	-	ns	
TDI/TMS hold after TCK rising edge	t <sub>7</sub> SR	6	-	-	ns	
TDO valid from TCK falling edge (propagation delay) <sup>2)</sup>	t <sub>8</sub> CC	-	32	36	ns	
TDO high impedance to valid output from TCK falling edge <sup>3)2)</sup>	t <sub>9</sub> CC	-	32	36	ns	
TDO valid output to high impedance from TCK falling edge <sup>2)</sup>	<i>t</i> <sub>10</sub> CC	-	32	36	ns	
TDO hold after TCK falling edge <sup>2)</sup>	<i>t</i> <sub>18</sub> CC	5	_	_	ns	

## Table 43 Interface Timing for Lower Voltage Range

1) The debug interface cannot operate faster than the overall system, therefore  $t_1 \ge t_{SYS}$ .

2) The falling edge on TCK is used to generate the TDO timing.

3) The setup time for TDO is given implicitly by the TCK cycle time.









