Infineon Technologies - C161SL25MAAFXUMA1 Datasheet

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	Obsolete
Core Processor	C166
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	63
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	PG-MQFP-80-7
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/c161sl25maafxuma1

2.2 Pin Configuration and Definition

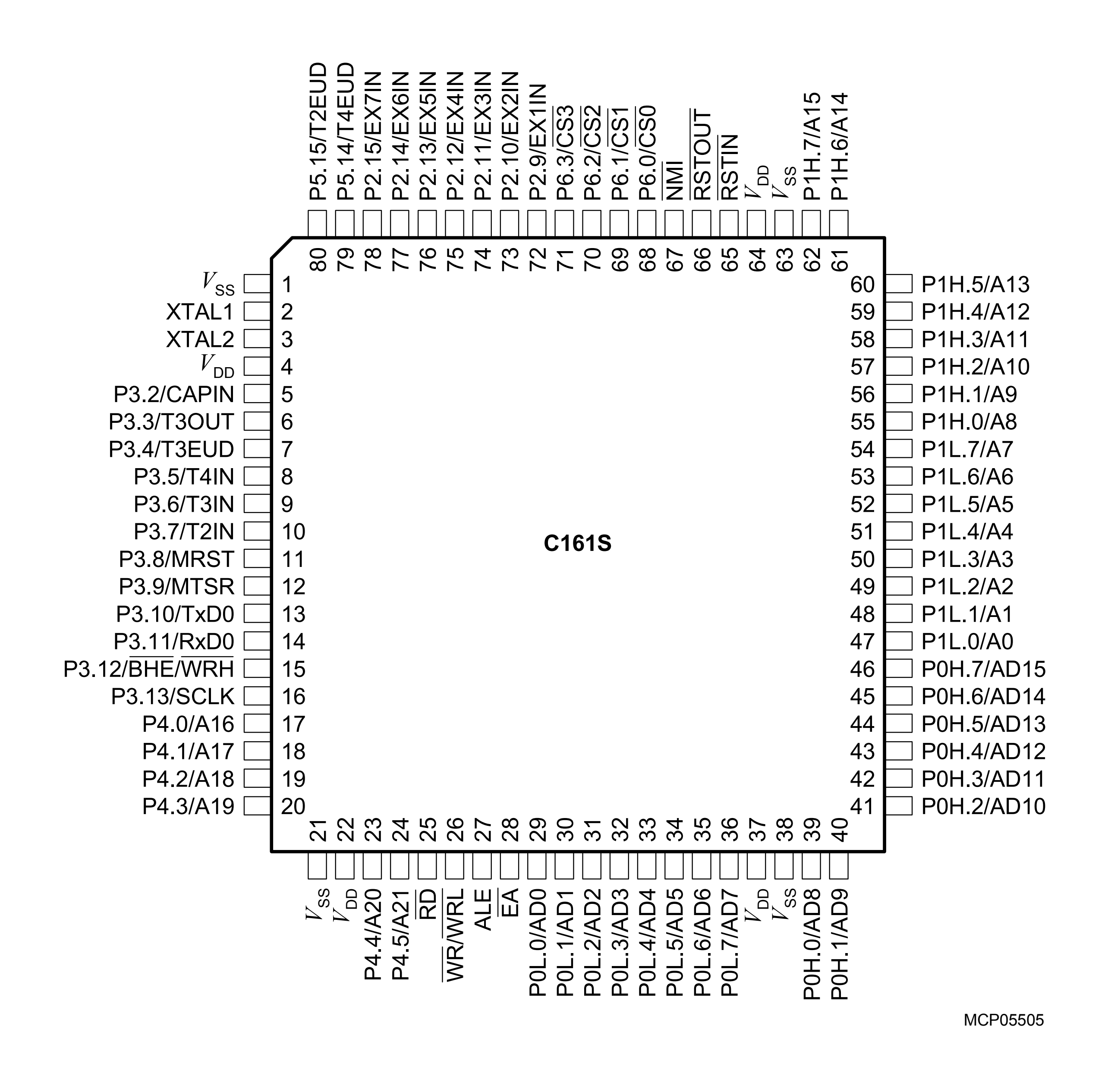


Figure 2 Pin Configuration (top view)

Table 2 **Pin Definitions and Functions** (cont'd)

Symbol	Pin No.	Input Outp.	Function
$\overline{\text{RD}}$	25	O	External Memory Read Strobe. $\overline{\text{RD}}$ is activated for every external instruction or data read access.
$\overline{\text{WR}}$/ $\overline{\text{WRL}}$	26	O	External Memory Write Strobe. In $\overline{\text{WR}}$-mode this pin is activated for every external data write access. In $\overline{\text{WRL}}$-mode this pin is activated for low byte data write accesses on a 16-bit bus, and for every data write access on an 8-bit bus. See WRCFG in register SYSCON for mode selection.
ALE	27	O	Address Latch Enable Output. Can be used for latching the address into external memory or an address latch in the multiplexed bus modes.
$\overline{\text{EA}}$	28	I	External Access Enable pin. A low level at this pin during and after Reset forces the C161S to begin instruction execution out of external memory. A high level forces execution out of the internal program memory. "ROMless" versions must have this pin tied to '0'.
PORT0 P0L.0-7 P0H.0-7	 29-36 39-46	IO	PORT0 consists of the two 8-bit bidirectional I/O ports P0L and P0H. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. In case of an external bus configuration, PORT0 serves as the address (A) and address/data (AD) bus in multiplexed bus modes and as the data (D) bus in demultiplexed bus modes. **Demultiplexed bus modes:** Data Path Width: 8-bit 16-bit P0L.0 – P0L.7: D0 – D7 D0 – D7 P0H.0 – P0H.7: I/O D8 – D15 **Multiplexed bus modes:** Data Path Width: 8-bit 16-bit P0L.0 – P0L.7: AD0 – AD7 AD0 – AD7 P0H.0 – P0H.7: A8 – A15 AD8 – AD15

3.2 External Bus Controller

All of the external memory accesses are performed by a particular on-chip External Bus Controller (EBC). It can be programmed either to Single Chip Mode when no external memory is required, or to one of four different external memory access modes, which are as follows:

- 16-/18-/20-/22-bit Addresses, 16-bit Data, Demultiplexed
- 16-/18-/20-/22-bit Addresses, 16-bit Data, Multiplexed
- 16-/18-/20-/22-bit Addresses, 8-bit Data, Multiplexed
- 16-/18-/20-/22-bit Addresses, 8-bit Data, Demultiplexed

In the demultiplexed bus modes, addresses are output on PORT1 and data is input/output on PORT0 or P0L, respectively. In the multiplexed bus modes both addresses and data use PORT0 for input/output.

Important timing characteristics of the external bus interface (Memory Cycle Time, Memory Tri-State Time, Length of ALE and Read Write Delay) have been made programmable to allow the user the adaption of a wide range of different types of memories and external peripherals.
In addition, up to 4 independent address windows may be defined (via register pairs ADDRSELx / BUSCONx) which control the access to different resources with different bus characteristics. These address windows are arranged hierarchically where BUSCON4 overrides BUSCON3 and BUSCON2 overrides BUSCON1. All accesses to locations not covered by these 4 address windows are controlled by BUSCON0.

Up to 4 external $\overline{\text{CS}}$ signals (3 windows plus default) can be generated in order to save external glue logic. The C161S offers the possibility to switch the $\overline{\text{CS}}$ outputs to an unlatched mode. In this mode the internal filter logic is switched off and the $\overline{\text{CS}}$ signals are directly generated from the address. The unlatched $\overline{\text{CS}}$ mode is enabled by setting CSCFG (SYSCON.6).

For applications which require less than 4 Mbytes of external memory space, this address space can be restricted to 1 Mbyte, 256 Kbytes, or to 64 Kbytes. In this case Port 4 outputs four, two, or no address lines at all. It outputs all 6 address lines, if the full address width is used.

3.3 Central Processing Unit (CPU)

The main core of the CPU consists of a 4-stage instruction pipeline, a 16-bit arithmetic and logic unit (ALU) and dedicated SFRs. Additional hardware has been spent for a separate multiply and divide unit, a bit-mask generator and a barrel shifter.

Based on these hardware provisions, most of the C161S's instructions can be executed in just one machine cycle which requires 100 ns at 20 MHz CPU clock. For example, shift and rotate instructions are always processed during one machine cycle independent of the number of bits to be shifted. All multiple-cycle instructions have been optimized so that they can be executed very fast as well: branches in 2 cycles, a 16 × 16 bit multiplication in 5 cycles and a 32-/16-bit division in 10 cycles. Another pipeline optimization, the so-called 'Jump Cache', allows reducing the execution time of repeatedly performed jumps in a loop from 2 cycles to 1 cycle.

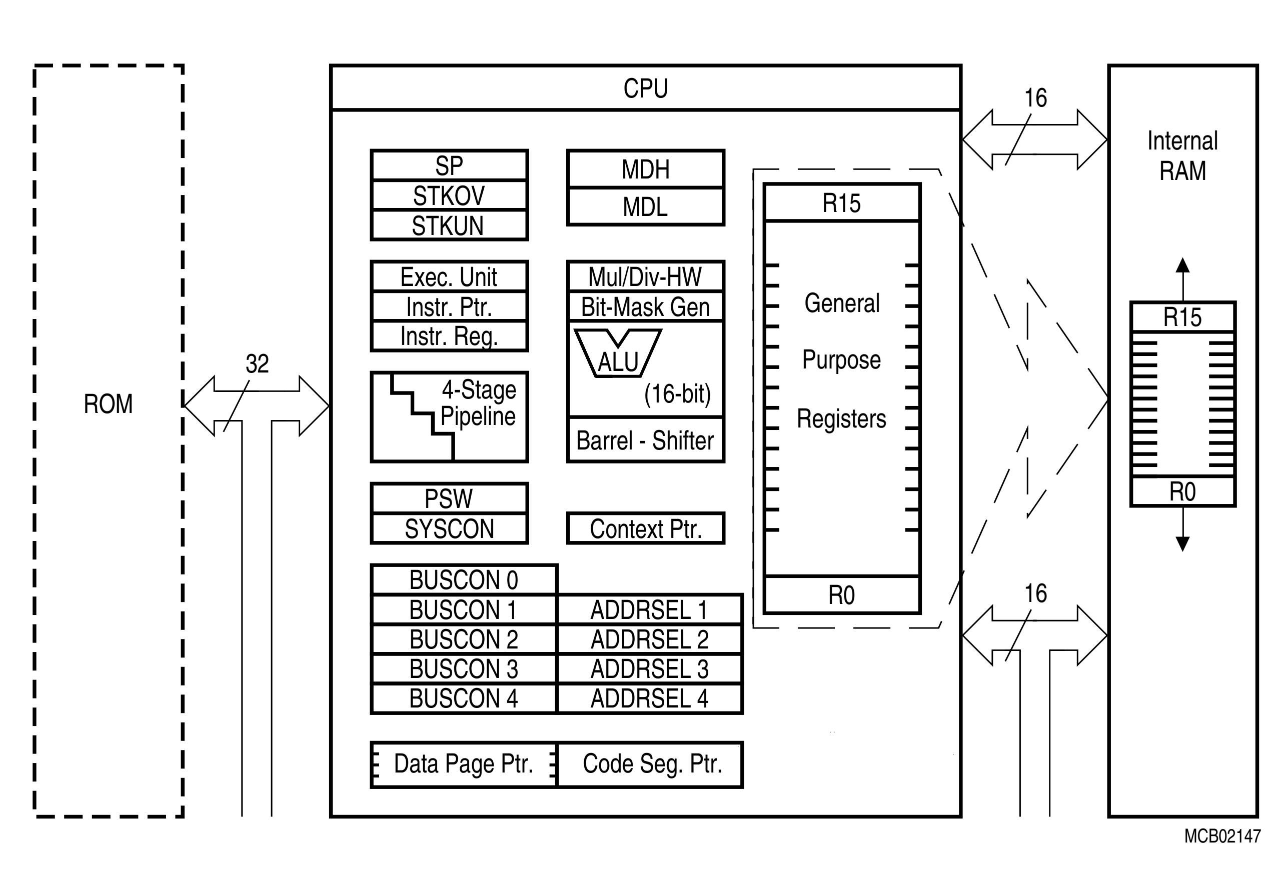


Figure 4 CPU Block Diagram

The CPU has a register context consisting of up to 16 wordwide GPRs at its disposal. These 16 GPRs are physically allocated within the on-chip RAM area. A Context Pointer (CP) register determines the base address of the active register bank to be accessed by the CPU at any time. The number of register banks 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 1024 words is provided as a storage for temporary data. The system stack is allocated in the on-chip RAM area, and it is accessed by the CPU via the stack pointer (SP) register. Two separate SFRs, STKOV and STKUN, are implicitly compared against the stack pointer value upon each stack access for the detection of a stack overflow or underflow.

The high performance offered by the hardware implementation of the CPU can efficiently be utilized by a programmer via the highly efficient C161S instruction set which includes the following instruction classes:

- 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.

The C161S also provides an excellent mechanism to identify and to process exceptions or error conditions that arise during run-time, so-called ‘Hardware Traps’. Hardware traps cause immediate non-maskable system reaction which is similar to a standard interrupt service (branching to a dedicated vector table location). The occurrence of a hardware trap is additionally signified by an individual bit in the trap flag register (TFR). Except when another higher prioritized trap service is in progress, a hardware trap will interrupt any actual program execution. In turn, hardware trap services can normally not be interrupted by standard or PEC interrupts.

Table 4 shows all of the possible exceptions or error conditions that can arise during run-time:

Table 4 Hardware Trap Summary

Exception Condition	Trap Flag	Trap Vector	Vector Location	Trap Number	Trap Priority
Reset Functions:	–				
• Hardware Reset		RESET	00’0000$_H$	00$_H$	III
• Software Reset		RESET	00’0000$_H$	00$_H$	III
• W-dog Timer Overflow		RESET	00’0000$_H$	00$_H$	III
Class A Hardware Traps:					
• Non-Maskable Interrupt	NMI	NMITRAP	00’0008$_H$	02$_H$	II
• Stack Overflow	STKOF	STOTRAP	00’0010$_H$	04$_H$	II
• Stack Underflow	STKUF	STUTRAP	00’0018$_H$	06$_H$	II
Class B Hardware Traps:					
• Undefined Opcode	UNDOPC	BTRAP	00’0028$_H$	0A$_H$	I
• Protected Instruction Fault	PRTFLT	BTRAP	00’0028$_H$	0A$_H$	I
• Illegal Word Operand Access	ILLOPA	BTRAP	00’0028$_H$	0A$_H$	I
• Illegal Instruction Access	ILLINA	BTRAP	00’0028$_H$	0A$_H$	I
• Illegal External Bus Access	ILLBUS	BTRAP	00’0028$_H$	0A$_H$	I
Reserved	–	–	[2C$_H$ – 3C$_H$]	[0B$_H$ – 0F$_H$]	–
Software Traps • TRAP Instruction	–	–	Any [00’0000$_H$ – 00’01FC$_H$] in steps of 4$_H$	Any [00$_H$ – 7F$_H$]	Current CPU Priority

3.4 General Purpose Timer (GPT) Unit

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

The GPT unit incorporates five 16-bit timers which are organized in two separate modules, GPT1 and GPT2. Each timer in each module may operate independently in a number of different modes, or may 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, which are Timer, Gated Timer, Counter, and Incremental Interface Mode. In Timer Mode, the input clock for a timer is derived from the CPU clock, divided by a programmable prescaler, while Counter Mode allows a timer to be clocked 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 gate or clock input. The maximum resolution of the timers in module GPT1 is 16 TCL.

The count direction (up/down) for each timer is programmable by software or may additionally be altered dynamically by an external signal on a port pin (TxEUD) to facilitate e.g. position tracking.

In Incremental Interface Mode the GPT1 timers (T2, T3, T4) can be directly connected to the incremental position sensor signals A and B via their respective inputs TxIN and TxEUD. Direction and count signals are internally derived from these two input signals, so 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, or may be used internally to clock timers T2 and T4 for measuring long time periods with high resolution.

In addition to their basic operating modes, timers T2 and T4 may be configured as reload or capture registers for timer T3. When used as capture or reload registers, timers T2 and T4 are stopped. The contents of timer T3 is captured into T2 or T4 in response to a signal at their associated input pins (TxIN). Timer T3 is reloaded with the contents of T2 or T4 triggered either by an external signal or by 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 constantly generated without software intervention.

3.5 Real Time Clock

The Real Time Clock (RTC) module of the C161S consists of a chain of 3 divider blocks, a fixed 8:1 divider, the reloadable 16-bit timer T14, and the 32-bit RTC timer (accessible via registers RTCH and RTCL). The RTC module is directly clocked with the on-chip oscillator frequency divided by 32 via a separate clock driver ($f_{RTC} = f_{OSC}/32$) and is therefore independent from the selected clock generation mode of the C161S. All timers count up.

The RTC module can be used for different purposes:

- System clock to determine the current time and date
- Cyclic time based interrupt
- 48-bit timer for long term measurements

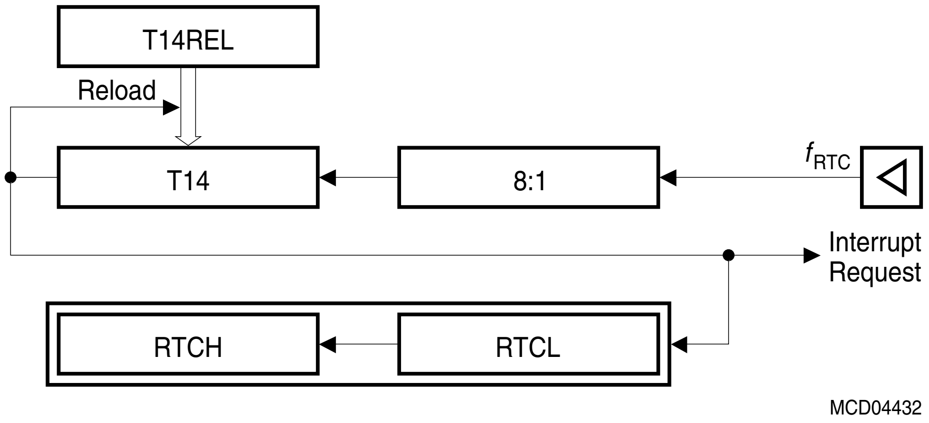


Figure 7 RTC Block Diagram

Note: The registers associated with the RTC are not affected by a reset in order to maintain the correct system time even when intermediate resets are executed.

3.6 Serial Channels

Serial communication with other microcontrollers, processors, terminals or external peripheral components is provided by two serial interfaces with different functionality, an Asynchronous/Synchronous Serial Channel (**ASC0**) and a High-Speed Synchronous Serial Channel (**SSC**).

The ASC0 is upward compatible with the serial ports of the Infineon 8-bit microcontroller families and supports full-duplex asynchronous communication at up to 625 kbit/s and half-duplex synchronous communication at up to 2.5 Mbit/s (@ 20 MHz CPU clock).
A dedicated baud rate generator allows to set up all standard baud rates without oscillator tuning. For transmission, reception and error handling 4 separate interrupt vectors are provided. In asynchronous mode, 8- or 9-bit data frames are transmitted or received, preceded by a start bit and terminated by one or two stop bits. For multiprocessor communication, a mechanism to distinguish address from data bytes has been included (8-bit data plus wake-up bit mode).
In synchronous mode, the ASC0 transmits or receives bytes (8 bits) synchronously to a shift clock which is generated by the ASC0. The ASC0 always shifts the LSB first. A loop back option is available for testing purposes.
A number of optional hardware error detection capabilities has been included to increase the reliability of data transfers. A parity bit can automatically be generated on transmission or be checked on reception. Framing error detection allows to recognize data frames with missing stop bits. An overrun error will be generated, if the last character received has not been read out of the receive buffer register at the time the reception of a new character is complete.

The SSC supports full-duplex synchronous communication at up to 5 Mbit/s (@ 20 MHz CPU clock). It may be configured so it interfaces with serially linked peripheral components. A dedicated baud rate generator allows to set up all standard baud rates without oscillator tuning. For transmission, reception, and error handling three separate interrupt vectors are provided.
The SSC transmits or receives characters of 2 … 16 bits length synchronously to a shift clock which can be generated by the SSC (master mode) or by an external master (slave mode). The SSC can start shifting with the LSB or with the MSB and allows the selection of shifting and latching clock edges as well as the clock polarity.
A number of optional hardware error detection capabilities has been included to increase the reliability of data transfers. Transmit and receive error supervise the correct handling of the data buffer. Phase and baudrate error detect incorrect serial data.

3.9 Oscillator Watchdog

The Oscillator Watchdog (OWD) monitors the clock signal generated by the on-chip oscillator (either with a crystal or via external clock drive). For this operation the PLL provides a clock signal which is used to supervise transitions on the oscillator clock. This PLL clock is independent from the XTAL1 clock. When the expected oscillator clock transitions are missing the OWD activates the PLL Unlock / OWD interrupt node and supplies the CPU with the PLL clock signal. Under these circumstances the PLL will oscillate with its basic frequency.

In direct drive mode the PLL base frequency is used directly (f_{CPU} = 2 … 5 MHz).
In prescaler mode the PLL base frequency is divided by 2 (f_{CPU} = 1 … 2.5 MHz).

Note: The CPU clock source is only switched back to the oscillator clock after a hardware reset.

The oscillator watchdog can be disabled by setting bit OWDDIS in register SYSCON. In this case (OWDDIS = ‘1’) the PLL remains idle and provides no clock signal, while the CPU clock signal is derived directly from the oscillator clock or via prescaler or SDD. Also no interrupt request will be generated in case of a missing oscillator clock.

Note: At the end of a reset bit OWDDIS reflects the inverted level of pin $\overline{\text{RD}}$ at that time. Thus the oscillator watchdog may also be disabled via hardware by (externally) pulling the $\overline{\text{RD}}$ line low upon a reset, similar to the standard reset configuration via PORT0.

3.10 Power Management

The C161S provides several means to control the power it consumes either at a given time or averaged over a certain timespan. Three mechanisms can be used (partly in parallel):

- **Power Saving Modes** switch the C161S into a special operating mode (control via instructions).
  Idle Mode stops the CPU while the peripherals can continue to operate.
  Power Down Mode stops all clock signals and all operation (RTC may optionally continue running).
- **Clock Generation Management** controls the distribution and the frequency of internal and external clock signals (control via register SYSCON2).
  Slow Down Mode lets the C161S run at a CPU clock frequency of f_{OSC} / 1 … 32 (half for prescaler operation) which drastically reduces the consumed power. The PLL can be optionally disabled while operating in Slow Down Mode.
- **Peripheral Management** permits temporary disabling of peripheral modules (control via register SYSCON3).
  Each peripheral can separately be disabled/enabled. A group control option disables a major part of the peripheral set by setting one single bit.

The on-chip RTC supports intermittent operation of the C161S by generating cyclic wake-up signals. This offers full performance to quickly react on action requests while the intermittent sleep phases greatly reduce the average power consumption of the system.

Table 5 Instruction Set Summary (cont'd)

Mnemonic	**Description**	**Bytes**
MOV(B)	Move word (byte) data	2 / 4
MOVBS	Move byte operand to word operand with sign extension	2 / 4
MOVBZ	Move byte operand to word operand with zero extension	2 / 4
JMPA, JMPI, JMPR	Jump absolute/indirect/relative if condition is met	4
JMPS	Jump absolute to a code segment	4
J(N)B	Jump relative if direct bit is (not) set	4
JBC	Jump relative and clear bit if direct bit is set	4
JNBS	Jump relative and set bit if direct bit is not set	4
CALLA, CALLI, CALLR	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	Return from intra-segment subroutine	2
RETS	Return from inter-segment subroutine	2
RETP	Return from intra-segment subroutine and pop direct word register from system stack	2
RETI	Return from interrupt service subroutine	2
SRST	Software Reset	4
IDLE	Enter Idle Mode	4
PWRDN	Enter Power Down Mode (supposes $\overline{\text{NMI}}$-pin being low)	4
SRVWDT	Service Watchdog Timer	4
DISWDT	Disable Watchdog Timer	4
EINIT	Signify End-of-Initialization on $\overline{\text{RSTOUT}}$-pin	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
NOP	Null operation	2

Table 6 C161S Registers, Ordered by Name (cont'd)

Name		Physical Address		8-Bit Addr.	Description	Reset Value
CP		FE10$_H$		08$_H$	CPU Context Pointer Register	FC00$_H$
CRIC	**b**	FF6A$_H$		B5$_H$	GPT2 CAPREL Interrupt Ctrl. Reg.	0000$_H$
CSP		FE08$_H$		04$_H$	CPU Code Seg. Pointer Reg. (read only)	0000$_H$
DP0H	**b**	F102$_H$	**E**	81$_H$	P0H Direction Control Register	00$_H$
DP0L	**b**	F100$_H$	**E**	80$_H$	P0L Direction Control Register	00$_H$
DP1H	**b**	F106$_H$	**E**	83$_H$	P1H Direction Control Register	00$_H$
DP1L	**b**	F104$_H$	**E**	82$_H$	P1L Direction Control Register	00$_H$
DP2	**b**	FFC2$_H$		E1$_H$	Port 2 Direction Control Register	0000$_H$
DP3	**b**	FFC6$_H$		E3$_H$	Port 3 Direction Control Register	0000$_H$
DP4	**b**	FFCA$_H$		E5$_H$	Port 4 Direction Control Register	00$_H$
DP6	**b**	FFCE$_H$		E7$_H$	Port 6 Direction Control Register	00$_H$
DPP0		FE00$_H$		00$_H$	CPU Data Page Pointer 0 Reg. (10 bits)	0000$_H$
DPP1		FE02$_H$		01$_H$	CPU Data Page Pointer 1 Reg. (10 bits)	0001$_H$
DPP2		FE04$_H$		02$_H$	CPU Data Page Pointer 2 Reg. (10 bits)	0002$_H$
DPP3		FE06$_H$		03$_H$	CPU Data Page Pointer 3 Reg. (10 bits)	0003$_H$
EXICON	**b**	F1C0$_H$	**E**	E0$_H$	External Interrupt Control Register	0000$_H$
IDCHIP		F07C$_H$	**E**	3E$_H$	Identifier	05XX$_H$
IDMANUF		F07E$_H$	**E**	3F$_H$	Identifier	1820$_H$
IDMEM		F07A$_H$	**E**	3D$_H$	Identifier	0000$_H$
IDMEM2		F076$_H$	**E**	3B$_H$	Identifier	0000$_H$
IDPROG		F078$_H$	**E**	3C$_H$	Identifier	0000$_H$
ISNC	**b**	F1DE$_H$	**E**	EF$_H$	Interrupt Subnode Control Register	0000$_H$
MDC	**b**	FF0E$_H$		87$_H$	CPU Multiply Divide Control Register	0000$_H$
MDH		FE0C$_H$		06$_H$	CPU Multiply Divide Reg. – High Word	0000$_H$
MDL		FE0E$_H$		07$_H$	CPU Multiply Divide Reg. – Low Word	0000$_H$
ODP2	**b**	F1C2$_H$	**E**	E1$_H$	Port 2 Open Drain Control Register	0000$_H$
ODP3	**b**	F1C6$_H$	**E**	E3$_H$	Port 3 Open Drain Control Register	0000$_H$
ODP6	**b**	F1CE$_H$	**E**	E7$_H$	Port 6 Open Drain Control Register	00$_H$
ONES	**b**	FF1E$_H$		8F$_H$	Constant Value 1's Register (read only)	FFFF$_H$
P0H	**b**	FF02$_H$		81$_H$	Port 0 High Reg. (Upper half of PORT0)	00$_H$

4.2 Operating Conditions

The following operating conditions must not be exceeded in order to ensure correct operation of the C161S. All parameters specified in the following sections refer to these operating conditions, unless otherwise noticed.

Table 8 Operating Condition Parameters

Parameter	Symbol	Limit Values		Unit	Notes
		Min.	Max.		
Standard digital supply voltage	V_{DD}	4.5	5.5	V	Active mode, f_{CPUmax} = 25 MHz
		2.5[1]	5.5	V	Power down mode
Reduced digital supply voltage	V_{DD}	3.0	3.6	V	Active mode, f_{CPUmax} = 20 MHz
		2.5[1]	3.6	V	Power down mode
Digital ground voltage	V_{SS}	0		V	Reference voltage
Overload current	I_{OV}	–	±5	mA	Per pin[2][3]
Absolute sum of overload currents	$\Sigma\|I_{OV}\|$	–	50	mA	[3]
External Load Capacitance	C_L	–	100	pF	–
Ambient temperature	T_A	0	70	°C	SAB-C161S …
		-40	85	°C	SAF-C161S …
		-40	125	°C	SAK-C161S …

[1] Output voltages and output currents will be reduced when V_{DD} leaves the range defined for active mode.

[2] Overload conditions occur if the standard operating conditions are exceeded, i.e. the voltage on any pin exceeds the specified range (i.e. $V_{OV} > V_{DD} + 0.5$ V or $V_{OV} < V_{SS} - 0.5$ V). The absolute sum of input overload currents on all pins may not exceed **50 mA**. The supply voltage must remain within the specified limits. Proper operation is not guaranteed if overload conditions occur on functional pins such as XTAL1, $\overline{\text{RD}}$, $\overline{\text{WR}}$, etc.

[3] Not subject to production test, verified by design/characterization.

Table 11 Power Consumption C161S (Standard Supply Voltage Range)
(Operating Conditions apply)

Parameter	Symbol	Limit Values		Unit	Test Condition
		Min.	Max.		
Power supply current (active) with all peripherals active	I_{DD5}	–	15 + 1.8 × f_{CPU}	mA	$\overline{\text{RSTIN}} = V_{IL2}$ f_{CPU} in [MHz][1]
Idle mode supply current with all peripherals active	I_{IDX5}	–	3 + 0.6 × f_{CPU}	mA	$\overline{\text{RSTIN}} = V_{IH1}$ f_{CPU} in [MHz][1]
Idle mode supply current with all peripherals deactivated, PLL off, SDD factor = 32	I_{IDO5}[2]	–	500 + 50 × f_{OSC}	µA	$\overline{\text{RSTIN}} = V_{IH1}$ f_{OSC} in [MHz][1]
Sleep and Power down mode supply current with RTC running	I_{PDR5}[2]	–	200 + 25 × f_{OSC}	µA	$V_{DD} = V_{DDmax}$ f_{OSC} in [MHz][3]
Sleep and Power down mode supply current with RTC disabled	I_{PDO5}	–	50	µA	$V_{DD} = V_{DDmax}$[3]

[1] The supply current is a function of the operating frequency. This dependency is illustrated in **Figure 8**. These parameters are tested at V_{DDmax} and maximum CPU clock with all outputs disconnected and all inputs at V_{IL} or V_{IH}.

[2] This parameter is determined mainly by the current consumed by the oscillator (see **Figure 9**). This current, however, is influenced by the external oscillator circuitry (crystal, capacitors). The values given refer to a typical circuitry and may change in case of a not optimized external oscillator circuitry.

[3] This parameter is tested including leakage currents. All inputs (including pins configured as inputs) at 0 V to 0.1 V or at V_{DD} - 0.1 V to V_{DD}, V_{REF} = 0 V, all outputs (including pins configured as outputs) disconnected.

Memory Cycle Variables

The timing tables below use three variables which are derived from the BUSCONx registers and represent the special characteristics of the programmed memory cycle. Table 15 describes, how these variables are to be computed.

Table 15 Memory Cycle Variables

Description	Symbol	Values
ALE Extension	t_A	TCL × <ALECTL>
Memory Cycle Time Waitstates	t_C	2TCL × (15 - <MCTC>)
Memory Tristate Time	t_F	2TCL × (1 - <MTTC>)

Note: Please respect the maximum operating frequency of the respective derivative.

5.4 AC Characteristics

Table 16 Multiplexed Bus (Standard Supply Voltage Range)
(Operating Conditions apply)

ALE cycle time = 6 TCL + $2t_A$ + t_C + t_F (120 ns at 25 MHz CPU clock without waitstates)

Parameter	Symbol		Max. CPU Clock = 25 MHz		Variable CPU Clock 1 / 2TCL = 1 to 25 MHz		Unit
			Min.	Max.	Min.	Max.	
ALE high time	t_5	CC	10 + t_A	–	TCL - 10 + t_A	–	ns
Address setup to ALE	t_6	CC	4 + t_A	–	TCL - 16 + t_A	–	ns
Address hold after ALE	t_7	CC	10 + t_A	–	TCL - 10 + t_A	–	ns
ALE falling edge to $\overline{\text{RD}}$, $\overline{\text{WR}}$ (with RW-delay)	t_8	CC	10 + t_A	–	TCL - 10 + t_A	–	ns
ALE falling edge to $\overline{\text{RD}}$, $\overline{\text{WR}}$ (no RW-delay)	t_9	CC	-10 + t_A	–	-10 + t_A	–	ns
Address float after $\overline{\text{RD}}$, $\overline{\text{WR}}$ (with RW-delay)	t_{10}	CC	–	6	–	6	ns
Address float after $\overline{\text{RD}}$, $\overline{\text{WR}}$ (no RW-delay)	t_{11}	CC	–	26	–	TCL + 6	ns

Table 16 Multiplexed Bus (Standard Supply Voltage Range) (cont'd)
(Operating Conditions apply)

ALE cycle time = 6 TCL + 2t_A + t_C + t_F (120 ns at 25 MHz CPU clock without waitstates)

Parameter	**Symbol**		**Max. CPU Clock = 25 MHz**		**Variable CPU Clock 1 / 2TCL = 1 to 25 MHz**		**Unit**
			Min.	**Max.**	**Min.**	**Max.**	
ALE fall. edge to $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ (with RW delay)	t_{42}	CC	16 + t_A	–	TCL - 4 + t_A	–	ns
ALE fall. edge to $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ (no RW delay)	t_{43}	CC	-4 + t_A	–	-4 + t_A	–	ns
Address float after $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ (with RW delay)	t_{44}	CC	–	0	–	0	ns
Address float after $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ (no RW delay)	t_{45}	CC	–	20	–	TCL	ns
$\overline{\text{RdCS}}$ to Valid Data In (with RW delay)	t_{46}	SR	–	16 + t_C	–	2TCL - 24 + t_C	ns
$\overline{\text{RdCS}}$ to Valid Data In (no RW delay)	t_{47}	SR	–	36 + t_C	–	3TCL - 24 + t_C	ns
$\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ Low Time (with RW delay)	t_{48}	CC	30 + t_C	–	2TCL - 10 + t_C	–	ns
$\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ Low Time (no RW delay)	t_{49}	CC	50 + t_C	–	3TCL - 10 + t_C	–	ns
Data valid to $\overline{\text{WrCS}}$	t_{50}	CC	26 + t_C	–	2TCL - 14 + t_C	–	ns
Data hold after $\overline{\text{RdCS}}$	t_{51}	SR	0	–	0	–	ns
Data float after $\overline{\text{RdCS}}$	t_{52}	SR	–	20 + t_F	–	2TCL - 20 + t_F	ns
Address hold after $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$	t_{54}	CC	20 + t_F	–	2TCL - 20 + t_F	–	ns
Data hold after $\overline{\text{WrCS}}$	t_{56}	CC	20 + t_F	–	2TCL - 20 + t_F	–	ns

[1] These parameters refer to the latched chip select signals ($\overline{\text{CSxL}}$). The early chip select signals ($\overline{\text{CSxE}}$) are specified together with the address and signal $\overline{\text{BHE}}$ (see figures below).

Table 17 **Multiplexed Bus (Reduced Supply Voltage Range)**
(Operating Conditions apply)

ALE cycle time = 6 TCL + $2t_A$ + t_C + t_F (150 ns at 20 MHz CPU clock without waitstates)

Parameter	**Symbol**		**Max. CPU Clock = 20 MHz**		**Variable CPU Clock 1 / 2TCL = 1 to 20 MHz**		**Unit**
			Min.	**Max.**	**Min.**	**Max.**	
ALE high time	t_5	CC	11 + t_A	–	TCL - 14 + t_A	–	ns
Address setup to ALE	t_6	CC	5 + t_A	–	TCL - 20 + t_A	–	ns
Address hold after ALE	t_7	CC	15 + t_A	–	TCL - 10 + t_A	–	ns
ALE falling edge to $\overline{\text{RD}}$, $\overline{\text{WR}}$ (with RW-delay)	t_8	CC	15 + t_A	–	TCL - 10 + t_A	–	ns
ALE falling edge to $\overline{\text{RD}}$, $\overline{\text{WR}}$ (no RW-delay)	t_9	CC	-10 + t_A	–	-10 + t_A	–	ns
Address float after $\overline{\text{RD}}$, $\overline{\text{WR}}$ (with RW-delay)	t_{10}	CC	–	6	–	6	ns
Address float after $\overline{\text{RD}}$, $\overline{\text{WR}}$ (no RW-delay)	t_{11}	CC	–	31	–	TCL + 6	ns
$\overline{\text{RD}}$, $\overline{\text{WR}}$ low time (with RW-delay)	t_{12}	CC	34 + t_C	–	2TCL - 16 + t_C	–	ns
$\overline{\text{RD}}$, $\overline{\text{WR}}$ low time (no RW-delay)	t_{13}	CC	59 + t_C	–	3TCL - 16 + t_C	–	ns
$\overline{\text{RD}}$ to valid data in (with RW-delay)	t_{14}	SR	–	22 + t_C	–	2TCL - 28 + t_C	ns
$\overline{\text{RD}}$ to valid data in (no RW-delay)	t_{15}	SR	–	47 + t_C	–	3TCL - 28 + t_C	ns
ALE low to valid data in	t_{16}	SR	–	45 + t_A + t_C	–	3TCL - 30 + t_A + t_C	ns
Address to valid data in	t_{17}	SR	–	57 + $2t_A$ + t_C	–	4TCL - 43 + $2t_A$ + t_C	ns
Data hold after $\overline{\text{RD}}$ rising edge	t_{18}	SR	0	–	0	–	ns
Data float after $\overline{\text{RD}}$	t_{19}	SR	–	36 + t_F	–	2TCL - 14 + t_F	ns

Table 18 Demultiplexed Bus (Standard Supply Voltage Range) (cont'd)
(Operating Conditions apply)

ALE cycle time = 4 TCL + $2t_A$ + t_C + t_F (80 ns at 25 MHz CPU clock without waitstates)

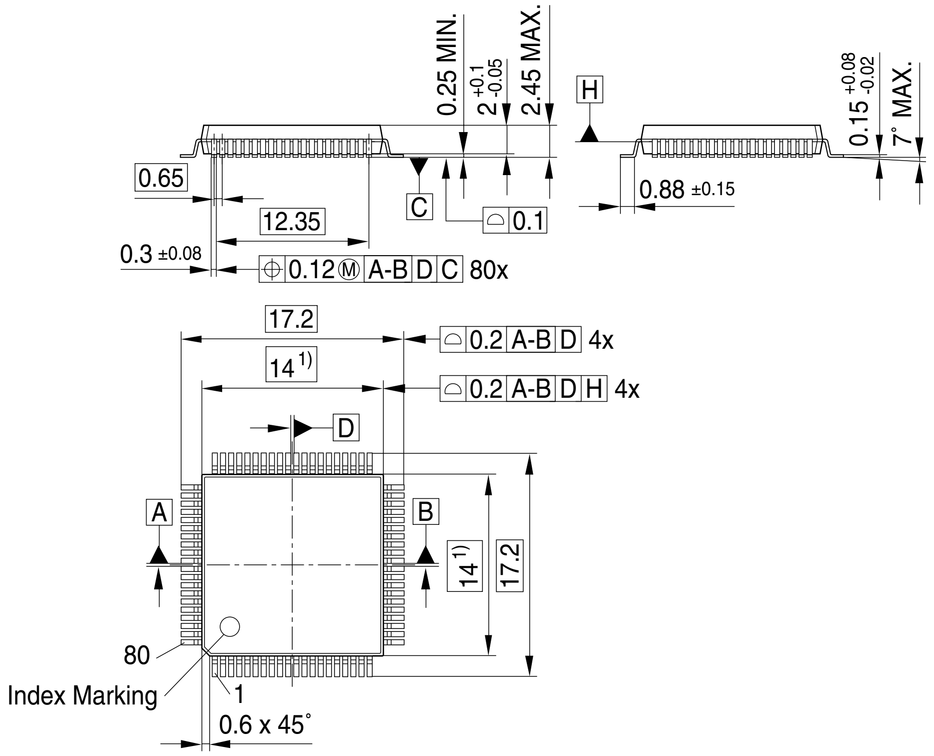
Parameter	Symbol		Max. CPU Clock = 25 MHz		Variable CPU Clock 1 / 2TCL = 1 to 25 MHz		Unit
			Min.	Max.	Min.	Max.	
Data float after $\overline{\text{RdCS}}$ (no RW-delay)[1]	t_{68}	SR	–	0 + t_F	–	TCL - 20 + $2t_A$ + t_F[1]	ns
Address hold after $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$	t_{55}	CC	-6 + t_F	–	-6 + t_F	–	ns
Data hold after $\overline{\text{WrCS}}$	t_{57}	CC	6 + t_F	–	TCL - 14 + t_F	–	ns

[1] RW-delay and t_A refer to the next following bus cycle (including an access to an on-chip X-Peripheral).

[2] Read data are latched with the same clock edge that triggers the address change and the rising $\overline{\text{RD}}$ edge. Therefore address changes before the end of $\overline{\text{RD}}$ have no impact on read cycles.

[3] These parameters refer to the latched chip select signals ($\overline{\text{CSxL}}$). The early chip select signals ($\overline{\text{CSxE}}$) are specified together with the address and signal $\overline{\text{BHE}}$ (see figures below).

6 Package Outlines



[1] Does not include plastic or metal protrusion of 0.25 max. per side

GPM05249

Figure 23 P-MQFP-80-7 (Plastic Metric Quad Flat Package)

You can find all of our packages, sorts of packing and others in our Infineon Internet Page "Products": **http://www.infineon.com/products**.

SMD = Surface Mounted Device

Dimensions in mm