Infineon Technologies - C164CI8EMDBKXUMA1 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	20MHz
Connectivity	CANbus, EBI/EMI, SPI, SSC, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	59
Program Memory Size	64KB (64K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°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/c164ci8emdbkxuma1

Data Sheet, V2.0, May 2001

C164CI/SI
C164CL/SL

16-Bit Single-Chip Microcontroller

Microcontrollers

Never stop thinking.

- Supported by a Large Range of Development Tools like C-Compilers, Macro-Assembler Packages, Emulators, Evaluation Boards, HLL-Debuggers, Simulators, Logic Analyzer Disassemblers, Programming Boards
- On-Chip Bootstrap Loader
- 80-Pin MQFP Package, 0.65 mm pitch

This document describes several derivatives of the C164 group. **Table 1** enumerates these derivatives and summarizes the differences. As this document refers to all of these derivatives, some descriptions may not apply to a specific product.

Table 1 C164CI Derivative Synopsis

Derivative[1)]	Program Memory	CAPCOM6	CAN Interf.	Operating Frequency
SAK-C164CI-8R[25]M SAF-C164CI-8R[25]M	64 KByte ROM	Full function	CAN1	20 MHz, [25 MHz]
SAK-C164SI-8R[25]M SAF-C164SI-8R[25]M	64 KByte ROM	Full function	---	20 MHz, [25 MHz]
SAK-C164CL-8R[25]M SAF-C164CL-8R[25]M	64 KByte ROM	Reduced fct.	CAN1	20 MHz, [25 MHz]
SAK-C164SL-8R[25]M SAF-C164SL-8R[25]M	64 KByte ROM	Reduced fct.	---	20 MHz, [25 MHz]
SAK-C164CL-6R[25]M SAF-C164CL-6R[25]M	48 KByte ROM	Reduced fct.	CAN1	20 MHz, [25 MHz]
SAK-C164SL-6R[25]M SAF-C164SL-6R[25]M	48 KByte ROM	Reduced fct.	---	20 MHz, [25 MHz]
SAK-C164CI-L[25]M SAF-C164CI-L[25]M	---	Full function	CAN1	20 MHz, [25 MHz]
SAK-C164CI-8EM SAF-C164CI-8EM	64 KByte OTP	Full function	CAN1	20 MHz

1) This Data Sheet is valid for ROM(less) devices starting with and including design step AB, and for OTP devices starting with and including design step DA.

For simplicity all versions are referred to by the term **C164CI** throughout this document.

Table 2 Pin Definitions and Functions (cont'd)

Symbol	Pin No.	Input Outp.	Function
XTAL2 XTAL1	54 55	O I	XTAL2: Output of the oscillator amplifier circuit. XTAL1: Input to the oscillator amplifier and input to the internal clock generator To clock the device from an external source, drive XTAL1, while leaving XTAL2 unconnected. Minimum and maximum high/low and rise/fall times specified in the AC Characteristics must be observed.
$\overline{\text{RSTIN}}$	69	I/O	Reset Input with Schmitt-Trigger characteristics. A low level at this pin while the oscillator is running resets the C164CI. An internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}. A spike filter suppresses input pulses <10 ns. Input pulses >100 ns safely pass the filter. The minimum duration for a safe recognition should be 100 ns + 2 CPU clock cycles. In bidirectional reset mode (enabled by setting bit BDRSTEN in register SYSCON) the $\overline{\text{RSTIN}}$ line is internally pulled low for the duration of the internal reset sequence upon any reset (HW, SW, WDT). See note below this table.
$\overline{\text{RST}}$ $\overline{\text{OUT}}$	70	O	Internal Reset Indication Output. This pin is set to a low level when the part is executing either a hardware-, a software- or a watchdog timer reset. $\overline{\text{RSTOUT}}$ remains low until the EINIT (end of initialization) instruction is executed.
$\overline{\text{NMI}}$	71	I	Non-Maskable Interrupt Input. A high to low transition at this pin causes the CPU to vector to the NMI trap routine. When the PWRDN (power down) instruction is executed, the $\overline{\text{NMI}}$ pin must be low in order to force the C164CI to go into power down mode. If $\overline{\text{NMI}}$ is high, when PWRDN is executed, the part will continue to run in normal mode. If not used, pin $\overline{\text{NMI}}$ should be pulled high externally.

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 C164CI's instructions can be executed in just one machine cycle which requires 2 CPU clocks (4 TCL). 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', reduces the execution time of repeatedly performed jumps in a loop from 2 cycles to 1 cycle.

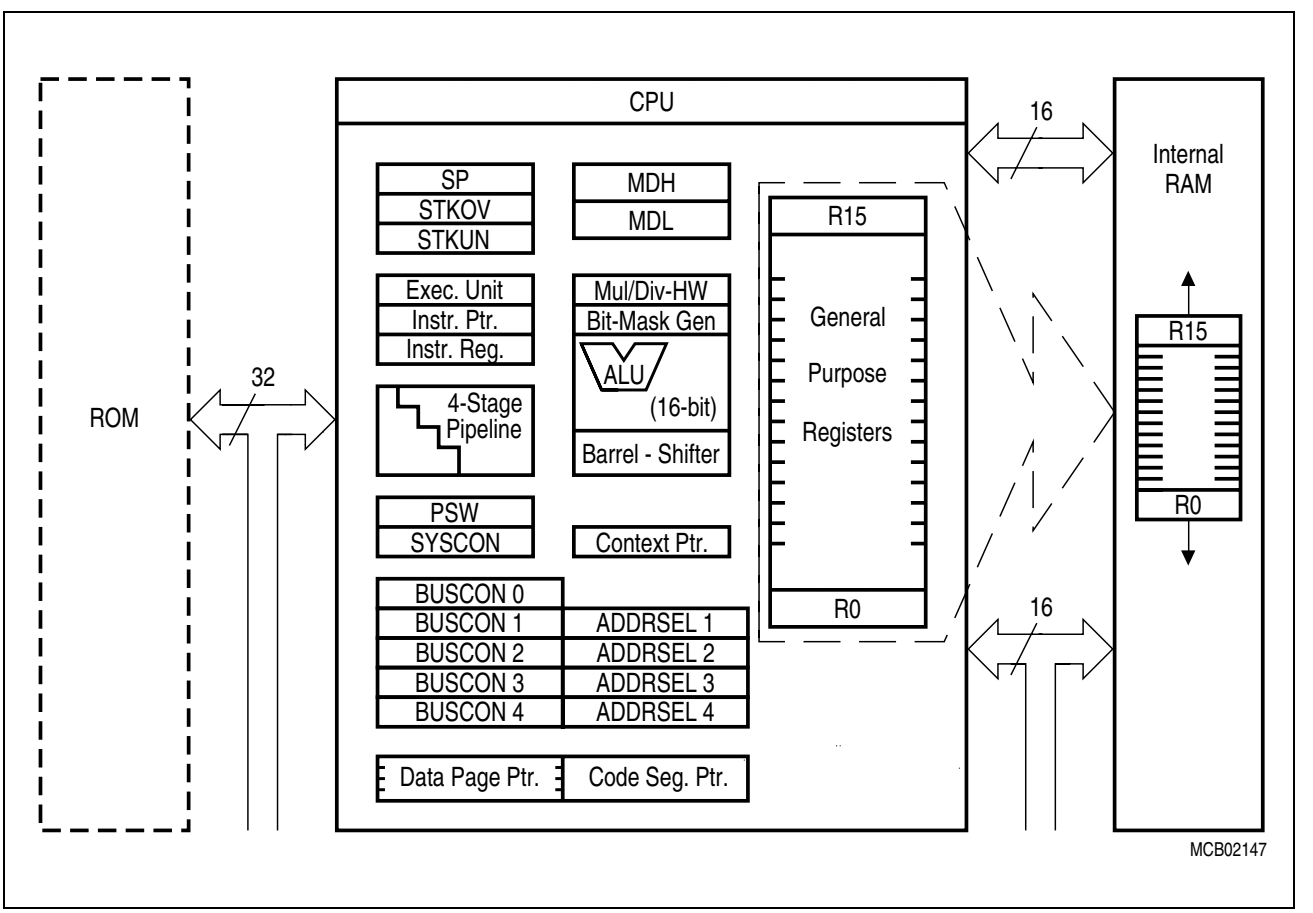

Figure 4 CPU Block Diagram

Interrupt System

With an interrupt response time within a range from just 5 to 12 CPU clocks (in case of internal program execution), the C164CI is capable of reacting very fast to the occurrence of non-deterministic events.

The architecture of the C164CI supports several mechanisms for fast and flexible response to service requests that can be generated from various sources internal or external to the microcontroller. Any of these interrupt requests can be programmed to being serviced by the Interrupt Controller or by the Peripheral Event Controller (PEC).

In contrast to a standard interrupt service where the current program execution is suspended and a branch to the interrupt vector table is performed, just one cycle is ‘stolen’ from the current CPU activity to perform a PEC service. A PEC service implies a single byte or word data transfer between any two memory locations with an additional increment of either the PEC source or the destination pointer. An individual PEC transfer counter is implicity decremented for each PEC service except when performing in the continuous transfer mode. When this counter reaches zero, a standard interrupt is performed to the corresponding source related vector location. PEC services are very well suited, for example, for supporting the transmission or reception of blocks of data. The C164CI has 8 PEC channels each of which offers such fast interrupt-driven data transfer capabilities.

A separate control register which contains an interrupt request flag, an interrupt enable flag and an interrupt priority bitfield exists for each of the possible interrupt sources. Via its related register, each source can be programmed to one of sixteen interrupt priority levels. Once having been accepted by the CPU, an interrupt service can only be interrupted by a higher prioritized service request. For the standard interrupt processing, each of the possible interrupt sources has a dedicated vector location.

Fast external interrupt inputs are provided to service external interrupts with high precision requirements. These fast interrupt inputs feature programmable edge detection (rising edge, falling edge or both edges).

Software interrupts are supported by means of the ‘TRAP’ instruction in combination with an individual trap (interrupt) number.

Table 3 shows all of the possible C164CI interrupt sources and the corresponding hardware-related interrupt flags, vectors, vector locations and trap (interrupt) numbers.

Note: Interrupt nodes which are not used by associated peripherals, may be used to generate software controlled interrupt requests by setting the respective interrupt request bit (xIR).

Table 3 C164CI Interrupt Nodes

Source of Interrupt or PEC Service Request	Request Flag	Enable Flag	Interrupt Vector	Vector Location	Trap Number
Fast External Interrupt 0	CC8IR	CC8IE	CC8INT	$00'0060_H$	18_H
Fast External Interrupt 1	CC9IR	CC9IE	CC9INT	$00'0064_H$	19_H
Fast External Interrupt 2	CC10IR	CC10IE	CC10INT	$00'0068_H$	$1A_H$
Fast External Interrupt 3	CC11IR	CC11IE	CC11INT	$00'006C_H$	$1B_H$
GPT1 Timer 2	T2IR	T2IE	T2INT	$00'0088_H$	22_H
GPT1 Timer 3	T3IR	T3IE	T3INT	$00'008C_H$	23_H
GPT1 Timer 4	T4IR	T4IE	T4INT	$00'0090_H$	24_H
A/D Conversion Complete	ADCIR	ADCIE	ADCINT	$00'00A0_H$	28_H
A/D Overrun Error	ADEIR	ADEIE	ADEINT	$00'00A4_H$	29_H
ASC0 Transmit	S0TIR	S0TIE	S0TINT	$00'00A8_H$	$2A_H$
ASC0 Transmit Buffer	S0TBIR	S0TBIE	S0TBINT	$00'011C_H$	47_H
ASC0 Receive	S0RIR	S0RIE	S0RINT	$00'00AC_H$	$2B_H$
ASC0 Error	S0EIR	S0EIE	S0EINT	$00'00B0_H$	$2C_H$
SSC Transmit	SCTIR	SCTIE	SCTINT	$00'00B4_H$	$2D_H$
SSC Receive	SCRIR	SCRIE	SCRINT	$00'00B8_H$	$2E_H$
SSC Error	SCEIR	SCEIE	SCEINT	$00'00BC_H$	$2F_H$
CAPCOM Register 16	CC16IR	CC16IE	CC16INT	$00'00C0_H$	30_H
CAPCOM Register 17	CC17IR	CC17IE	CC17INT	$00'00C4_H$	31_H
CAPCOM Register 18	CC18IR	CC18IE	CC18INT	$00'00C8_H$	32_H
CAPCOM Register 19	CC19IR	CC19IE	CC19INT	$00'00CC_H$	33_H
CAPCOM Register 24	CC24IR	CC24IE	CC24INT	$00'00E0_H$	38_H
CAPCOM Register 25	CC25IR	CC25IE	CC25INT	$00'00E4_H$	39_H
CAPCOM Register 26	CC26IR	CC26IE	CC26INT	$00'00E8_H$	$3A_H$
CAPCOM Register 27	CC27IR	CC27IE	CC27INT	$00'00EC_H$	$3B_H$
CAPCOM Timer 7	T7IR	T7IE	T7INT	$00'00F4_H$	$3D_H$
CAPCOM Timer 8	T8IR	T8IE	T8INT	$00'00F8_H$	$3E_H$
CAPCOM6 Interrupt	CC6IR	CC6IE	CC6INT	$00'00FC_H$	$3F_H$
CAN Interface 1	XP0IR	XP0IE	XP0INT	$00'0100_H$	40_H
PLL/OWD and RTC	XP3IR	XP3IE	XP3INT	$00'010C_H$	43_H

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 three 16-bit timers. Each timer may operate independently in a number of different modes, or may be concatenated with another timer.

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 over-flow/underflow. The state of this latch 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.

A/D Converter

For analog signal measurement, a 10-bit A/D converter with 8 multiplexed input channels and a sample and hold circuit has been integrated on-chip. It uses the method of successive approximation. The sample time (for loading the capacitors) and the conversion time is programmable and can so be adjusted to the external circuitry.

Overrun error detection/protection is provided for the conversion result register (ADDAT): either an interrupt request will be generated when the result of a previous conversion has not been read from the result register at the time the next conversion is complete, or the next conversion is suspended in such a case until the previous result has been read.

For applications which require less than 8 analog input channels, the remaining channel inputs can be used as digital input port pins.

The A/D converter of the C164CI supports four different conversion modes. In the standard Single Channel conversion mode, the analog level on a specified channel is sampled once and converted to a digital result. In the Single Channel Continuous mode, the analog level on a specified channel is repeatedly sampled and converted without software intervention. In the Auto Scan mode, the analog levels on a prespecified number of channels (standard or extension) are sequentially sampled and converted. In the Auto Scan Continuous mode, the number of prespecified channels is repeatedly sampled and converted. In addition, the conversion of a specific channel can be inserted (injected) into a running sequence without disturbing this sequence. This is called Channel Injection Mode.

The Peripheral Event Controller (PEC) may be used to automatically store the conversion results into a table in memory for later evaluation, without requiring the overhead of entering and exiting interrupt routines for each data transfer.

After each reset and also during normal operation the ADC automatically performs calibration cycles. This automatic self-calibration constantly adjusts the converter to changing operating conditions (e.g. temperature) and compensates process variations.

These calibration cycles are part of the conversion cycle, so they do not affect the normal operation of the A/D converter.

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 IO or input stages under software control. This can be selected for each pin separately via register P5DIDIS (Port 5 Digital Input Disable).

CAN-Module

The integrated CAN-Module handles the completely autonomous transmission and reception of CAN frames in accordance with the CAN specification V2.0 part B (active), i.e. the on-chip CAN-Modules can receive and transmit standard frames with 11-bit identifiers as well as extended frames with 29-bit identifiers.

The module provides Full CAN functionality on up to 15 message objects. Message object 15 may be configured for Basic CAN functionality. Both modes provide separate masks for acceptance filtering which allows to accept a number of identifiers in Full CAN mode and also allows to disregard a number of identifiers in Basic CAN mode. All message objects can be updated independent from the other objects and are equipped for the maximum message length of 8 bytes.

The bit timing is derived from the XCLK and is programmable up to a data rate of 1 Mbit/s. Each CAN-Module uses two pins of Port 4 or Port 8 to interface to an external bus transceiver. The interface pins are assigned via software.

Note: When the CAN interface is assigned to Port 4, the respective segment address lines on Port 4 cannot be used. This will limit the external address space.

Watchdog Timer

The Watchdog Timer represents one of the fail-safe mechanisms which have been implemented to prevent the controller from malfunctioning for longer periods of time.

The Watchdog Timer is always enabled after a reset of the chip, and can only be disabled in the time interval until the EINIT (end of initialization) instruction has been executed. Thus, the chip's start-up procedure is always monitored. The software has to be designed to service the Watchdog Timer before it overflows. If, due to hardware or software related failures, the software fails to do so, the Watchdog Timer overflows and generates an internal hardware reset and pulls the $\overline{\text{RSTOUT}}$ pin low in order to allow external hardware components to be reset.

The Watchdog Timer is a 16-bit timer, clocked with the system clock divided by 2/4/128/256. The high byte of the Watchdog Timer register can be set to a prespecified reload value (stored in WDTREL) in order to allow further variation of the monitored time interval. Each time it is serviced by the application software, the high byte of the Watchdog Timer is reloaded. Thus, time intervals between 20 µs and 336 ms can be monitored (@ 25 MHz).
The default Watchdog Timer interval after reset is 5.24 ms (@ 25 MHz).

Table 7 C164CI Registers, Ordered by Name (cont'd)

Name	Physical Address	8-Bit Addr.	Description	Reset Value
C1MCFGn	$EFn6_H$ **X**	---	CAN Message Configuration Register (msg. **n**)	UU_H
C1MCRn	$EFn0_H$ **X**	---	CAN Message Control Register (msg. **n**)	$UUUU_H$
C1PCIR	$EF02_H$ **X**	---	CAN1 Port Control / Interrupt Register	$XXXX_H$
C1UARn	$EFn2_H$ **X**	---	CAN Upper Arbitration Register (msg. **n**)	$UUUU_H$
C1UGML	$EF08_H$ **X**	---	CAN Upper Global Mask Long	$UUUU_H$
C1UMLM	$EF0C_H$ **X**	---	CAN Upper Mask of Last Message	$UUUU_H$
CC10IC b	$FF8C_H$	$C6_H$	External Interrupt 2 Control Register	0000_H
CC11IC b	$FF8E_H$	$C7_H$	External Interrupt 3 Control Register	0000_H
CC16	$FE60_H$	30_H	CAPCOM Register 16	0000_H
CC16IC b	$F160_H$ **E**	$B0_H$	CAPCOM Reg. 16 Interrupt Ctrl. Reg.	0000_H
CC17	$FE62_H$	31_H	CAPCOM Register 17	0000_H
CC17IC b	$F162_H$ **E**	$B1_H$	CAPCOM Reg. 17 Interrupt Ctrl. Reg.	0000_H
CC18	$FE64_H$	32_H	CAPCOM Register 18	0000_H
CC18IC b	$F164_H$ **E**	$B2_H$	CAPCOM Reg. 18 Interrupt Ctrl. Reg.	0000_H
CC19	$FE66_H$	33_H	CAPCOM Register 19	0000_H
CC19IC b	$F166_H$ **E**	$B3_H$	CAPCOM Reg. 19 Interrupt Ctrl. Reg.	0000_H
CC20	$FE68_H$	34_H	CAPCOM Register 20	0000_H
CC20IC b	$F168_H$ **E**	$B4_H$	CAPCOM Reg. 20 Interrupt Ctrl. Reg.	0000_H
CC21	$FE6A_H$	35_H	CAPCOM Register 21	0000_H
CC21IC b	$F16A_H$ **E**	$B5_H$	CAPCOM Reg. 21 Interrupt Ctrl. Reg.	0000_H
CC22	$FE6C_H$	36_H	CAPCOM Register 22	0000_H
CC22IC b	$F16C_H$ **E**	$B6_H$	CAPCOM Reg. 22 Interrupt Ctrl. Reg.	0000_H
CC23	$FE6E_H$	37_H	CAPCOM Register 23	0000_H
CC23IC b	$F16E_H$ **E**	$B7_H$	CAPCOM Reg. 23 Interrupt Ctrl. Reg.	0000_H
CC24	$FE70_H$	38_H	CAPCOM Register 24	0000_H
CC24IC b	$F170_H$ **E**	$B8_H$	CAPCOM Reg. 24 Interrupt Ctrl. Reg.	0000_H
CC25	$FE72_H$	39_H	CAPCOM Register 25	0000_H
CC25IC b	$F172_H$ **E**	$B9_H$	CAPCOM Reg. 25 Interrupt Ctrl. Reg.	0000_H
CC26	$FE74_H$	$3A_H$	CAPCOM Register 26	0000_H

Table 7 C164CI Registers, Ordered by Name (cont'd)

Name	Physical Address	8-Bit Addr.	Description	Reset Value
OPDAT	$EDC4_H$ **X**	---	OTP Progr. Interface Data Register	0000_H
P0H b	$FF02_H$	81_H	Port 0 High Reg. (Upper half of PORT0)	00_H
P0L b	$FF00_H$	80_H	Port 0 Low Reg. (Lower half of PORT0)	00_H
P1H b	$FF06_H$	83_H	Port 1 High Reg. (Upper half of PORT1)	00_H
P1L b	$FF04_H$	82_H	Port 1 Low Reg. (Lower half of PORT1)	00_H
P3 b	$FFC4_H$	$E2_H$	Port 3 Register	0000_H
P4 b	$FFC8_H$	$E4_H$	Port 4 Register (7 bits)	00_H
P5 b	$FFA2_H$	$D1_H$	Port 5 Register (read only)	$XXXX_H$
P5DIDIS b	$FFA4_H$	$D2_H$	Port 5 Digital Input Disable Register	0000_H
P8 b	$FFD4_H$	EA_H	Port 8 Register (8 bits)	00_H
PECC0	$FEC0_H$	60_H	PEC Channel 0 Control Register	0000_H
PECC1	$FEC2_H$	61_H	PEC Channel 1 Control Register	0000_H
PECC2	$FEC4_H$	62_H	PEC Channel 2 Control Register	0000_H
PECC3	$FEC6_H$	63_H	PEC Channel 3 Control Register	0000_H
PECC4	$FEC8_H$	64_H	PEC Channel 4 Control Register	0000_H
PECC5	$FECA_H$	65_H	PEC Channel 5 Control Register	0000_H
PECC6	$FECC_H$	66_H	PEC Channel 6 Control Register	0000_H
PECC7	$FECE_H$	67_H	PEC Channel 7 Control Register	0000_H
PICON b	$F1C4_H$ **E**	$E2_H$	Port Input Threshold Control Register	0000_H
POCON0H	$F082_H$ **E**	41_H	Port P0H Output Control Register	0011_H
POCON0L	$F080_H$ **E**	40_H	Port P0L Output Control Register	0011_H
POCON1H	$F086_H$ **E**	43_H	Port P1H Output Control Register	0011_H
POCON1L	$F084_H$ **E**	42_H	Port P1L Output Control Register	0011_H
POCON20	$F0AA_H$ **E**	55_H	Dedicated Pin Output Control Register	0000_H
POCON3	$F08A_H$ **E**	45_H	Port P3 Output Control Register	2222_H
POCON4	$F08C_H$ **E**	46_H	Port P4 Output Control Register	0010_H
POCON8	$F092_H$ **E**	49_H	Port P8 Output Control Register	0022_H
PSW b	$FF10_H$	88_H	CPU Program Status Word	0000_H
RP0H b	$F108_H$ **E**	84_H	System Startup Config. Reg. (Rd. only)	XX_H
RSTCON b	$F1E0_H$ **m**	---	Reset Control Register	$00XX_H$

Operating Conditions

The following operating conditions must not be exceeded in order to ensure correct operation of the C164CI. All parameters specified in the following sections refer to these operating conditions, unless otherwise noticed.

Table 9 Operating Condition Parameters

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Digital supply voltage	V_{DD}	4.75	5.5	V	Active mode, f_{CPUmax} = 25 MHz
		2.5[1]	5.5	V	PowerDown 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	Pin drivers in **default** mode[4][5]
Ambient temperature	T_A	0	70	°C	SAB-C164CI …
		-40	85	°C	SAF-C164CI …
		-40	125	°C	SAK-C164CI …

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 operatings 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 line XTAL1, $\overline{RD}$, $\overline{WR}$, etc.

3) Not 100% tested, guaranteed by design and characterization.

4) The timing is valid for pin drivers operating in default current mode (selected after reset). Reducing the output current may lead to increased delays or reduced driving capability (C_L).

5) The current ROM-version of the C164CI is equipped with port drivers, which provide reduced driving capability and reduced control. Please refer to the actual errata sheet for details.

Table 10 Current Limits for Port Output Drivers

Port Output Driver Mode	Maximum Output Current (I_{OLmax}, $-I_{OHmax}$)[1]	Nominal Output Current (I_{OLnom}, $-I_{OHnom}$)[2]
Strong driver	10 mA	2.5 mA
Medium driver	4.0 mA	1.0 mA
Weak driver	0.5 mA	0.1 mA

1) An output current above $|I_{OXnom}|$ may be drawn from up to three pins at the same time. For any group of 16 neighboring port output pins the total output current in each direction (ΣI_{OL} and $\Sigma -I_{OH}$) must remain below 50 mA.

2) The current ROM-version of the C164CI (step Ax) is equipped with port drivers, which provide reduced driving capability and reduced control. Please refer to the actual errata sheet for details.

Power Consumption C164CI (ROM)

(Operating Conditions apply)

Parameter	Sym-bol	Limit Values		Unit	Test Conditions
		min.	max.		
Power supply current (active) with all peripherals active	I_{DD}	–	1 + 2.5 × f_{CPU}	mA	$\overline{RSTIN}$ = V_{IL} f_{CPU} in [MHz][1]
Idle mode supply current with all peripherals active	I_{IDX}	–	1 + 1.1 × f_{CPU}	mA	$\overline{RSTIN}$ = V_{IH1} f_{CPU} in [MHz][1]
Idle mode supply current with all peripherals deactivated, PLL off, SDD factor = 32	I_{IDO}[2]	–	500 + 50 × f_{OSC}	µA	$\overline{RSTIN}$ = V_{IH1} f_{OSC} in [MHz][1]
Sleep and Power-down mode supply current with RTC running	I_{PDR}[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_{PDO}	–	50	µA	V_{DD} = V_{DDmax}[3]

1) The supply current is a function of the operating frequency. This dependency is illustrated in **Figure 9**. 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 8**). 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.

8) During the sample time the input capacitance C_{AIN} can be charged/discharged by the external source. The internal resistance of the analog source must allow the capacitance to reach its final voltage level within t_S. After the end of the sample time t_S, changes of the analog input voltage have no effect on the conversion result. Values for the sample time t_S depend on programming and can be taken from **Table 14**.

Sample time and conversion time of the C164CI's A/D Converter are programmable. **Table 14** should be used to calculate the above timings.
The limit values for f_{BC} must not be exceeded when selecting ADCTC.

Table 14 A/D Converter Computation Table

ADCON.15|14 (ADCTC)	A/D Converter Basic Clock f_{BC}	ADCON.13|12 (ADSTC)	Sample time t_S
00	f_{CPU} / 4	00	$t_{BC} \times 8$
01	f_{CPU} / 2	01	$t_{BC} \times 16$
10	f_{CPU} / 16	10	$t_{BC} \times 32$
11	f_{CPU} / 8	11	$t_{BC} \times 64$

Converter Timing Example:

Assumptions: f_{CPU} = 25 MHz (i.e. t_{CPU} = 40 ns), ADCTC = '00', ADSTC = '00'.
Basic clock f_{BC} = f_{CPU}/4 = 6.25 MHz, i.e. t_{BC} = 160 ns.
Sample time t_S = $t_{BC} \times 8$ = 1280 ns.
Conversion time t_C = t_S + 40 t_{BC} + 2 t_{CPU} = (1280 + 6400 + 80) ns = 7.8 µs.

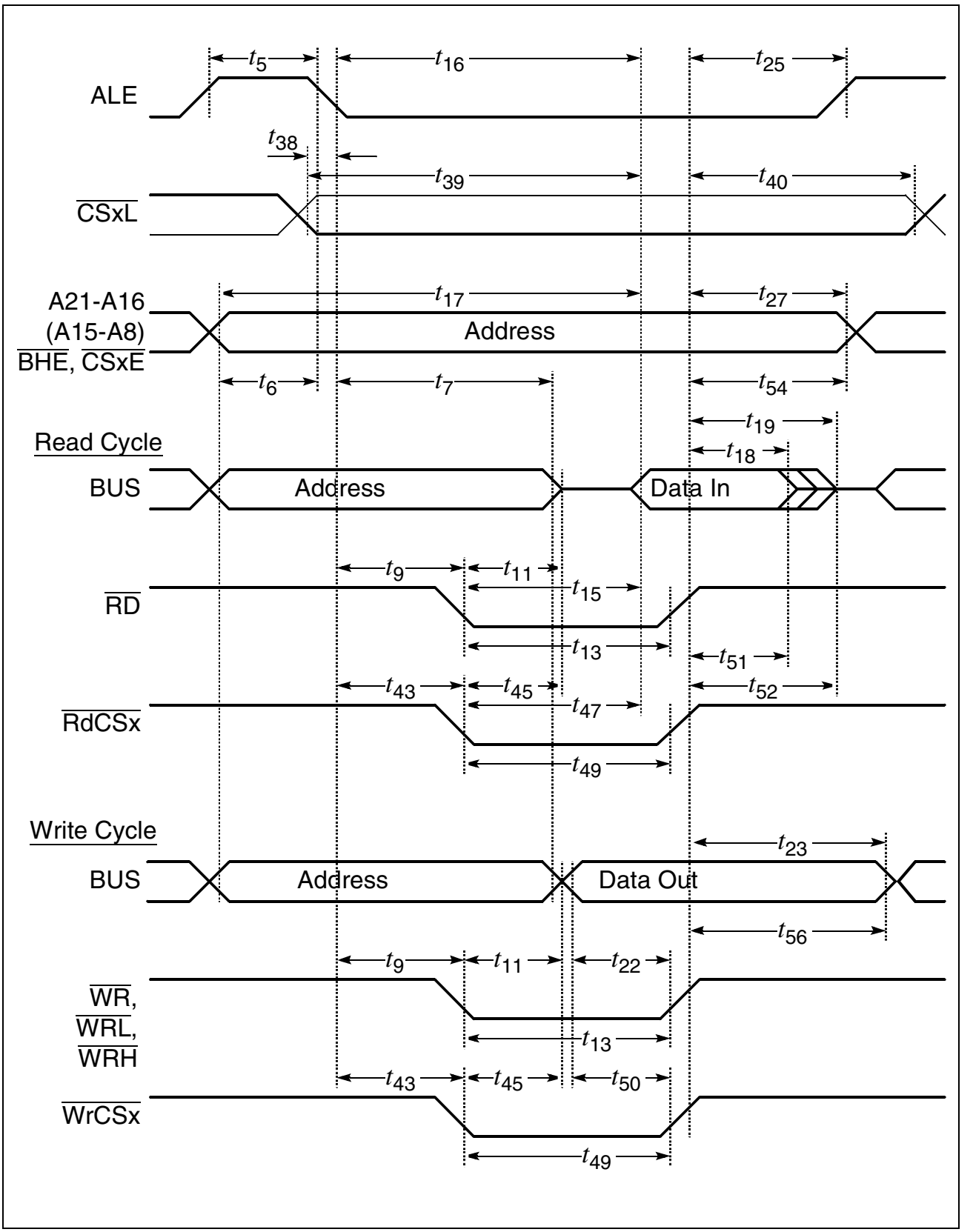

**Figure 19 External Memory Cycle:
Multiplexed Bus, No Read/Write Delay, Extended ALE**

Demultiplexed Bus (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 valid to $\overline{WR}$	t_{22}	CC	20 + t_C	–	2TCL - 20 + t_C	–	ns
Data hold after $\overline{WR}$	t_{24}	CC	10 + t_F	–	TCL - 10 + t_F	–	ns
ALE rising edge after $\overline{RD}$, $\overline{WR}$	t_{26}	CC	-10 + t_F	–	-10 + t_F	–	ns
Address hold after $\overline{WR}$[2]	t_{28}	CC	0 + t_F	–	0 + t_F	–	ns
ALE falling edge to $\overline{CS}$[3]	t_{38}	CC	-4 - t_A	10 - t_A	-4 - t_A	10 - t_A	ns
$\overline{CS}$ low to Valid Data In[3]	t_{39}	SR	–	40 + t_C + $2t_A$	–	3TCL - 20 + t_C + $2t_A$	ns
$\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$[3]	t_{41}	CC	6 + t_F	–	TCL - 14 + t_F	–	ns
ALE falling edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW-delay)	t_{42}	CC	16 + t_A	–	TCL - 4 + t_A	–	ns
ALE falling edge to $\overline{RdCS}$, $\overline{WrCS}$ (no RW-delay)	t_{43}	CC	-4 + t_A	–	-4 + t_A	–	ns
$\overline{RdCS}$ to Valid Data In (with RW-delay)	t_{46}	SR	–	16 + t_C	–	2TCL - 24 + t_C	ns
$\overline{RdCS}$ to Valid Data In (no RW-delay)	t_{47}	SR	–	36 + t_C	–	3TCL - 24 + t_C	ns
$\overline{RdCS}$, $\overline{WrCS}$ Low Time (with RW-delay)	t_{48}	CC	30 + t_C	–	2TCL - 10 + t_C	–	ns
$\overline{RdCS}$, $\overline{WrCS}$ Low Time (no RW-delay)	t_{49}	CC	50 + t_C	–	3TCL - 10 + t_C	–	ns
Data valid to $\overline{WrCS}$	t_{50}	CC	26 + t_C	–	2TCL - 14 + t_C	–	ns
Data hold after $\overline{RdCS}$	t_{51}	SR	0	–	0	–	ns
Data float after $\overline{RdCS}$ (with RW-delay)[1]	t_{53}	SR	–	20 + t_F	–	2TCL - 20 + $2t_A$ + t_F[1]	ns

Demultiplexed Bus (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).

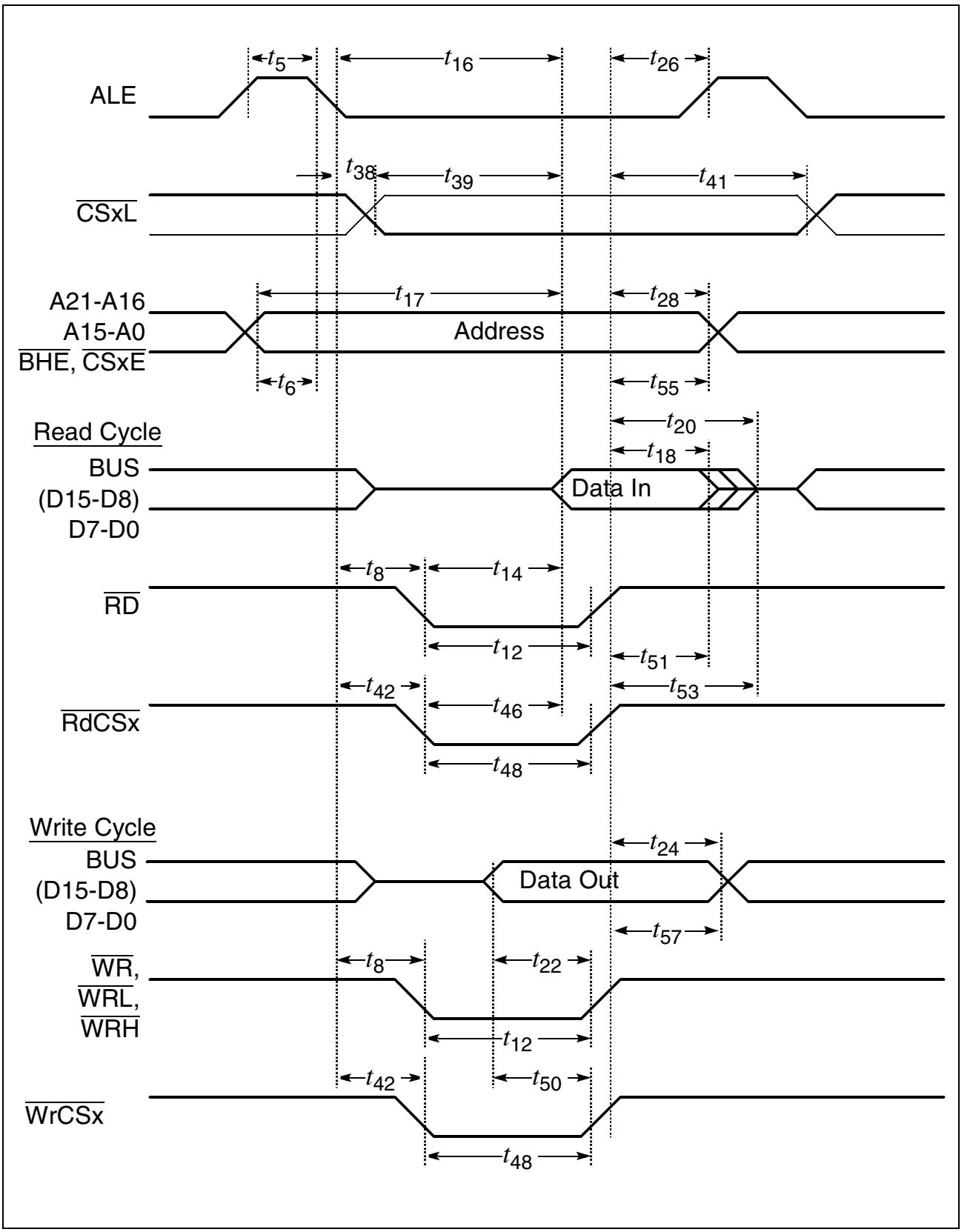

Figure 20 External Memory Cycle:
Demultiplexed Bus, With Read/Write Delay, Normal ALE

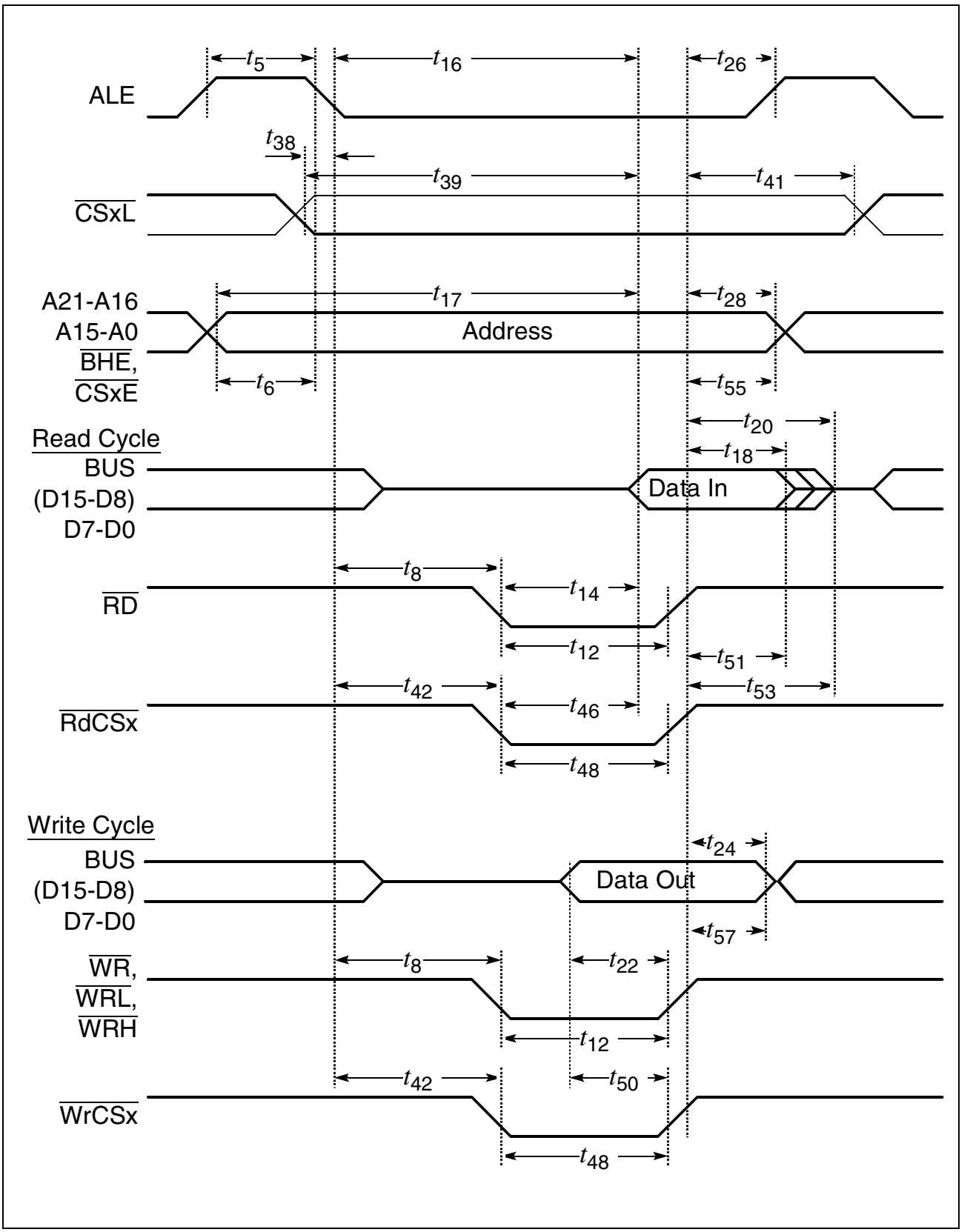

Figure 21 External Memory Cycle:
Demultiplexed Bus, With Read/Write Delay, Extended ALE

AC Characteristics

CLKOUT

(Operating Conditions apply)

Parameter	Symbol	Max. CPU Clock = 25 MHz		Variable CPU Clock 1 / 2TCL = 1 to 25 MHz		Unit
		min.	max.	min.	max.	
CLKOUT cycle time	t_{29} CC	40	40	2TCL	2TCL	ns
CLKOUT high time	t_{30} CC	14	–	TCL - 6	–	ns
CLKOUT low time	t_{31} CC	10	–	TCL - 10	–	ns
CLKOUT rise time	t_{32} CC	–	4	–	4	ns
CLKOUT fall time	t_{33} CC	–	4	–	4	ns
CLKOUT rising edge to ALE falling edge	t_{34} CC	$0 + t_A$	$10 + t_A$	$0 + t_A$	$10 + t_A$	ns

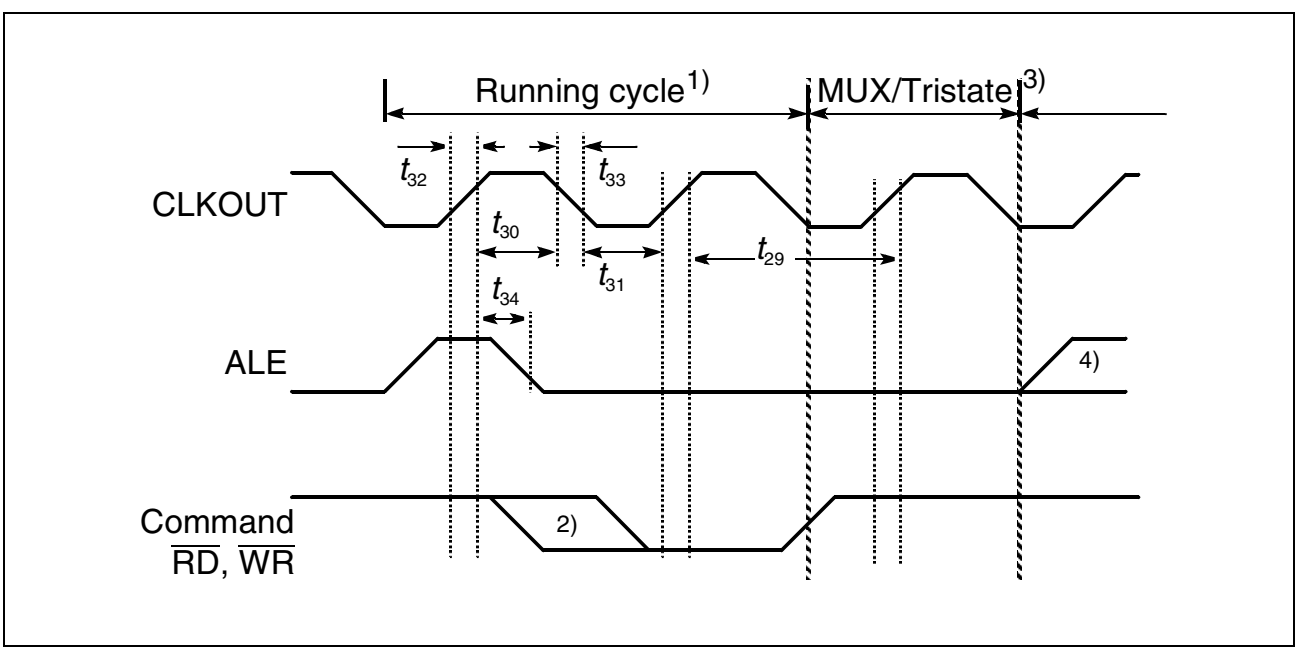

Figure 24 CLKOUT Timing

Notes

1) Cycle as programmed, including MCTC waitstates (Example shows 0 MCTC WS).
2) The leading edge of the respective command depends on RW-delay.
3) Multiplexed bus modes have a MUX waitstate added after a bus cycle, and an additional MTTC waitstate may be inserted here.
For a multiplexed bus with MTTC waitstate this delay is 2 CLKOUT cycles, for a demultiplexed bus without MTTC waitstate this delay is zero.
4) The next external bus cycle may start here.