Infineon Technologies - C164CI8E25MDBKXUMA1 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	Active
Core Processor	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/c164ci8e25mdbkxuma1

16-Bit Single-Chip Microcontroller C166 Family

C164CI

C164CI/SI, C164CL/SL

- High Performance 16-bit CPU with 4-Stage Pipeline
  - 80 ns Instruction Cycle Time at 25 MHz CPU Clock
  - 400 ns Multiplication (16 × 16 bit), 800 ns Division (32 / 16 bit)
  - Enhanced Boolean Bit Manipulation Facilities
  - Additional Instructions to Support HLL and Operating Systems
  - Register-Based Design with Multiple Variable Register Banks
  - Single-Cycle Context Switching Support
  - 16 MBytes Total Linear Address Space for Code and Data
  - 1024 Bytes On-Chip Special Function Register Area
- 16-Priority-Level Interrupt System with 32 Sources, Sample-Rate down to 40 ns
- 8-Channel Interrupt-Driven Single-Cycle Data Transfer Facilities via Peripheral Event Controller (PEC)
- Clock Generation via on-chip PLL (factors 1:1.5/2/2.5/3/4/5), via prescaler or via direct clock input
- On-Chip Memory Modules
  - 2 KBytes On-Chip Internal RAM (IRAM)
  - 2 KBytes On-Chip Extension RAM (XRAM)
  - up to 64 KBytes On-Chip Program Mask ROM or OTP Memory
- On-Chip Peripheral Modules
  - 8-Channel 10-bit A/D Converter with Programmable Conversion Time down to 7.8 µs
  - 8-Channel General Purpose Capture/Compare Unit (CAPCOM2)
  - Capture/Compare Unit for flexible PWM Signal Generation (CAPCOM6) (3/6 Capture/Compare Channels and 1 Compare Channel)
  - Multi-Functional General Purpose Timer Unit with 3 Timers
  - Two Serial Channels (Synchronous/Asynchronous and High-Speed-Synchronous)
  - On-Chip CAN Interface (Rev. 2.0B active) with 15 Message Objects (Full CAN/Basic CAN)
  - On-Chip Real Time Clock
- Up to 4 MBytes External Address Space for Code and Data
  - Programmable External Bus Characteristics for Different Address Ranges
  - Multiplexed or Demultiplexed External Address/Data Buses with 8-Bit or 16-Bit Data Bus Width
  - Four Optional Programmable Chip-Select Signals
- Idle, Sleep, and Power Down Modes with Flexible Power Management
- Programmable Watchdog Timer and Oscillator Watchdog
- Up to 59 General Purpose I/O Lines, partly with Selectable Input Thresholds and Hysteresis

Functional Description

The architecture of the C164CI combines advantages of both RISC and CISC processors and of advanced peripheral subsystems in a very well-balanced way. In addition the on-chip memory blocks allow the design of compact systems with maximum performance.
The following block diagram gives an overview of the different on-chip components and of the advanced, high bandwidth internal bus structure of the C164CI.

Note: All time specifications refer to a CPU clock of 25 MHz (see definition in the AC Characteristics section).

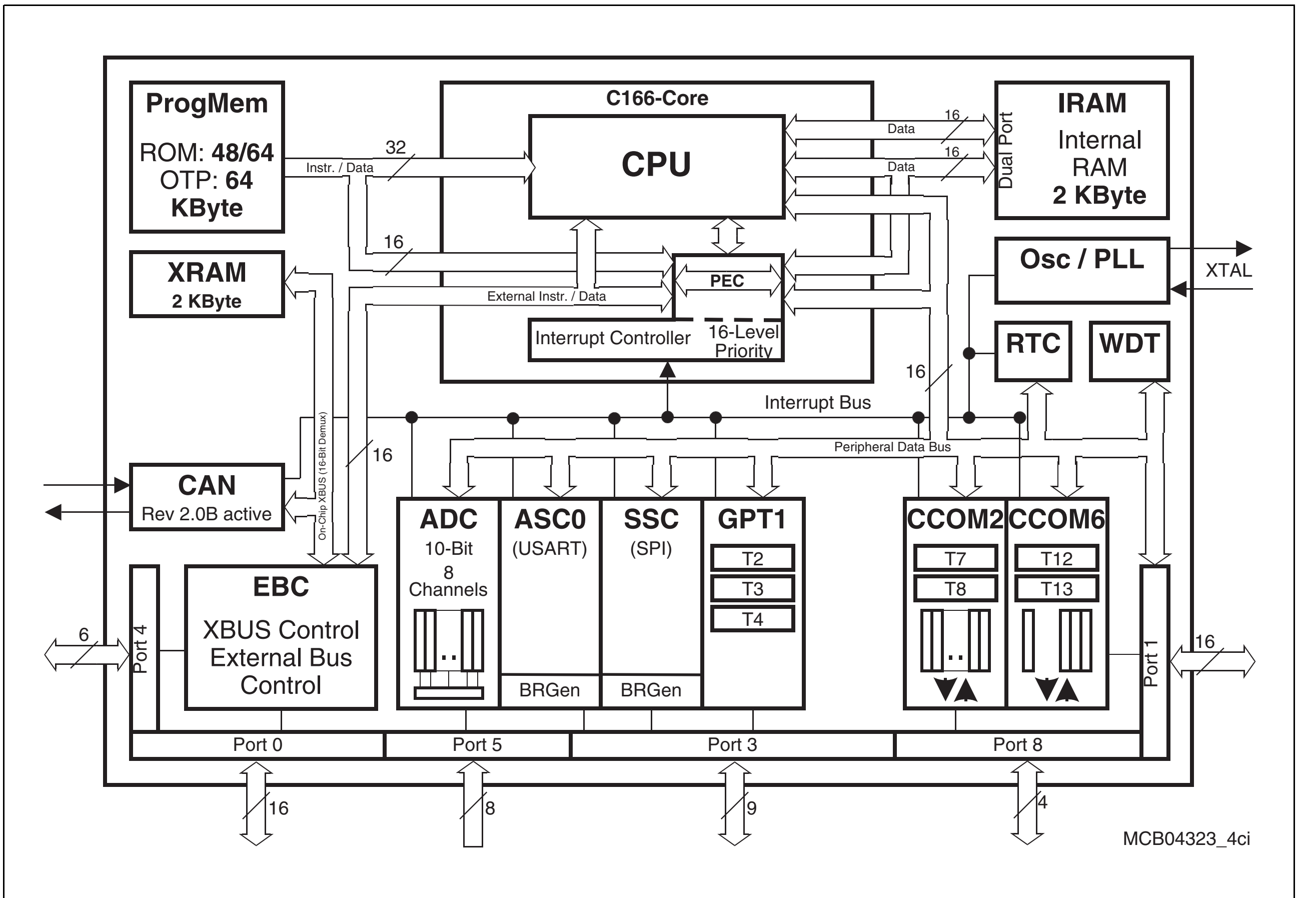


Figure 3 Block Diagram

The program memory, the internal RAM (IRAM) and the set of generic peripherals are connected to the CPU via separate buses. A fourth bus, the XBUS, connects external resources as well as additional on-chip resoures, the X-Peripherals (see Figure 3).

The XBUS resources (XRAM, CAN) of the C164CI can be enabled or disabled during initialization by setting the general X-Peripheral enable bit XPEN (SYSCON.2). Modules that are disabled consume neither address space nor port pins.

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

Table 3 C164CI Interrupt Nodes (cont'd)

Source of Interrupt or PEC Service Request	**Request Flag**	**Enable Flag**	**Interrupt Vector**	**Vector Location**	**Trap Number**
CAPCOM 6 Timer 12	T12IR	T12IE	T12INT	$00'0134_H$	$4D_H$
CAPCOM 6 Timer 13	T13IR	T13IE	T13INT	$00'0138_H$	$4E_H$
CAPCOM 6 Emergency	CC6EIR	CC6EIE	CC6EINT	$00'013C_H$	$4F_H$

The C164CI 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 occurence 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

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).

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 $\overline{RD}$ at that time. Thus the oscillator watchdog may also be disabled via hardware by (externally) pulling the $\overline{RD}$ line low upon a reset, similar to the standard reset configuration via PORT0.

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
CC26IC	**b**	$F174_H$	**E**	BA_H	CAPCOM Reg. 26 Interrupt Ctrl. Reg.	0000_H
CC27		$FE76_H$		$3B_H$	CAPCOM Register 27	0000_H
CC27IC	**b**	$F176_H$	**E**	BB_H	CAPCOM Reg. 27 Interrupt Ctrl. Reg.	0000_H
CC28		$FE78_H$		$3C_H$	CAPCOM Register 28	0000_H
CC28IC	**b**	$F178_H$	**E**	BC_H	CAPCOM Reg. 28 Interrupt Ctrl. Reg.	0000_H
CC29		$FE7A_H$		$3D_H$	CAPCOM Register 29	0000_H
CC29IC	**b**	$F184_H$	**E**	$C2_H$	CAPCOM Reg. 29 Interrupt Ctrl. Reg.	0000_H
CC30		$FE7C_H$		$3E_H$	CAPCOM Register 30	0000_H
CC30IC	**b**	$F18C_H$	**E**	$C6_H$	CAPCOM Reg. 30 Interrupt Ctrl. Reg.	0000_H
CC31		$FE7E_H$		$3F_H$	CAPCOM Register 31	0000_H
CC31IC	**b**	$F194_H$	**E**	CA_H	CAPCOM Reg. 31 Interrupt Ctrl. Reg.	0000_H
CC60		$FE30_H$		18_H	CAPCOM 6 Register 0	0000_H
CC61		$FE32_H$		19_H	CAPCOM 6 Register 1	0000_H
CC62		$FE34_H$		$1A_H$	CAPCOM 6 Register 2	0000_H
CC6EIC	**b**	$F188_H$	**E**	$C4_H$	CAPCOM 6 Emergency Interrrupt Control Register	0000_H
CC6CIC	**b**	$F17E_H$	**E**	BF_H	CAPCOM 6 Interrupt Control Register	0000_H
CC6MCON	**b**	$FF32_H$		99_H	CAPCOM 6 Mode Control Register	$00FF_H$
CC6MIC	**b**	$FF36_H$		$9B_H$	CAPCOM 6 Mode Interrupt Ctrl. Reg.	0000_H
CC6MSEL		$F036_H$	**E**	$1B_H$	CAPCOM 6 Mode Select Register	0000_H
CC8IC	**b**	$FF88_H$		$C4_H$	External Interrupt 0 Control Register	0000_H
CC9IC	**b**	$FF8A_H$		$C5_H$	External Interrupt 1 Control Register	0000_H
CCM4	**b**	$FF22_H$		91_H	CAPCOM Mode Control Register 4	0000_H
CCM5	**b**	$FF24_H$		92_H	CAPCOM Mode Control Register 5	0000_H
CCM6	**b**	$FF26_H$		93_H	CAPCOM Mode Control Register 6	0000_H
CCM7	**b**	$FF28_H$		94_H	CAPCOM Mode Control Register 7	0000_H
CMP13		$FE36_H$		$1B_H$	CAPCOM 6 Timer 13 Compare Reg.	0000_H
CP		$FE10_H$		08_H	CPU Context Pointer Register	$FC00_H$
CSP		$FE08_H$		04_H	CPU Code Segment Pointer Register (8 bits, not directly writeable)	0000_H

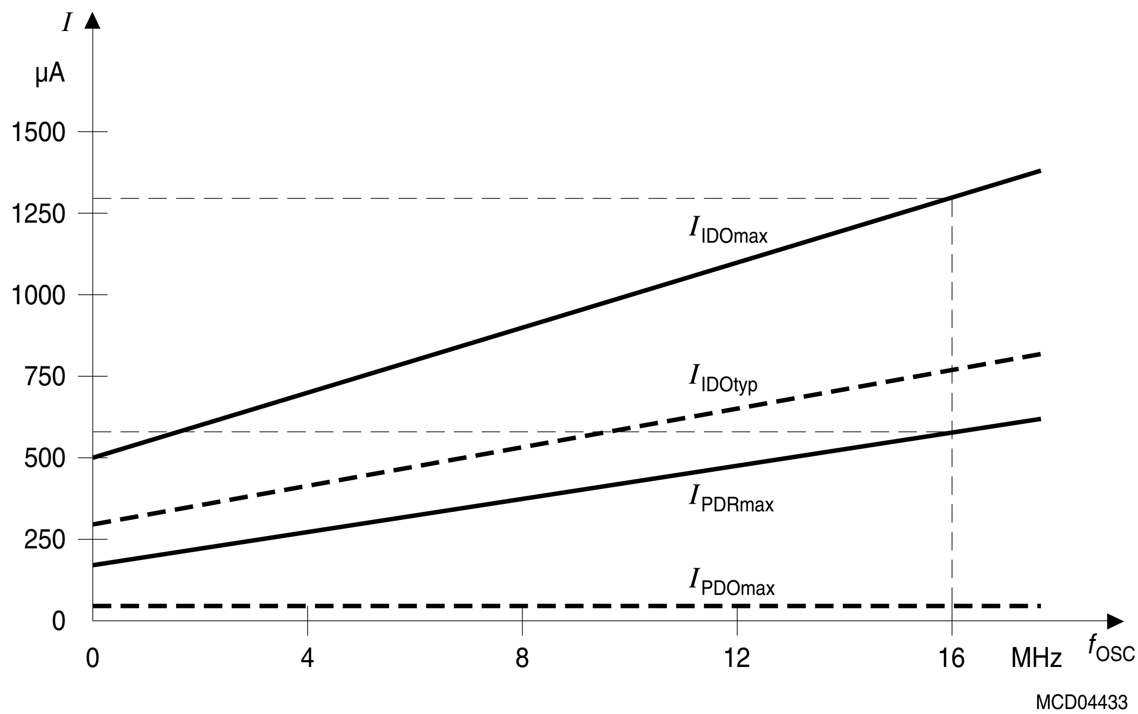


Figure 8 Idle and Power Down Supply Current as a Function of Oscillator Frequency

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.

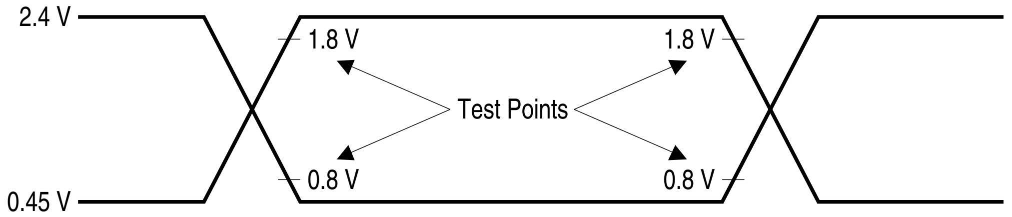
Table 14 A/D Converter Computation Table

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.

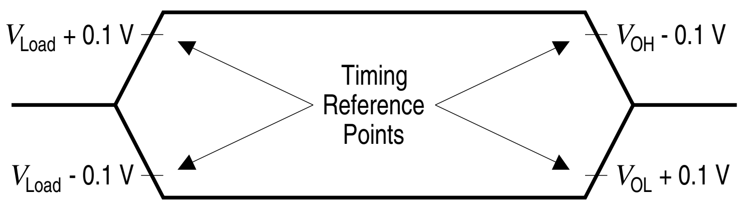
Testing Waveforms



AC inputs during testing are driven at 2.4 V for a logic '1' and 0.45 V for a logic '0'.
Timing measurements are made at V_{IH} min for a logic '1' and V_{IL} max for a logic '0'.

MCA04414

Figure 14 Input Output Waveforms



For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, but begins to float when a 100 mV change from the loaded V_{OH} / V_{OL} level occurs (I_{OH} / I_{OL} = 20 mA).

MCA00763

Figure 15 Float Waveforms

Multiplexed Bus (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.	
$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	t_{13}	CC	50 + t_C	–	3TCL - 10 + t_C	–	ns
$\overline{RD}$ to valid data in (with RW-delay)	t_{14}	SR	–	20 + t_C	–	2TCL - 20 + t_C	ns
$\overline{RD}$ to valid data in (no RW-delay)	t_{15}	SR	–	40 + t_C	–	3TCL - 20 + t_C	ns
ALE low to valid data in	t_{16}	SR	–	40 + t_A + t_C	–	3TCL - 20 + t_A + t_C	ns
Address to valid data in	t_{17}	SR	–	50 + $2t_A$ + t_C	–	4TCL - 30 + $2t_A$ + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}	SR	0	–	0	–	ns
Data float after $\overline{RD}$	t_{19}	SR	–	26 + t_F	–	2TCL - 14 + t_F	ns
Data valid to $\overline{WR}$	t_{22}	CC	20 + t_C	–	2TCL - 20 + t_C	–	ns
Data hold after $\overline{WR}$	t_{23}	CC	26 + t_F	–	2TCL - 14 + t_F	–	ns
ALE rising edge after $\overline{RD}$, $\overline{WR}$	t_{25}	CC	26 + t_F	–	2TCL - 14 + t_F	–	ns
Address hold after $\overline{RD}$, $\overline{WR}$	t_{27}	CC	26 + t_F	–	2TCL - 14 + t_F	–	ns
ALE falling edge to $\overline{CS}$[1)]	t_{38}	CC	-4 - t_A	10 - t_A	-4 - t_A	10 - t_A	ns
$\overline{CS}$ low to Valid Data In[1)]	t_{39}	SR	–	40 + t_C + $2t_A$	–	3TCL - 20 + t_C + $2t_A$	ns
$\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$[1)]	t_{40}	CC	46 + t_F	–	3TCL - 14 + t_F	–	ns
ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay)	t_{42}	CC	16 + t_A	–	TCL - 4 + t_A	–	ns

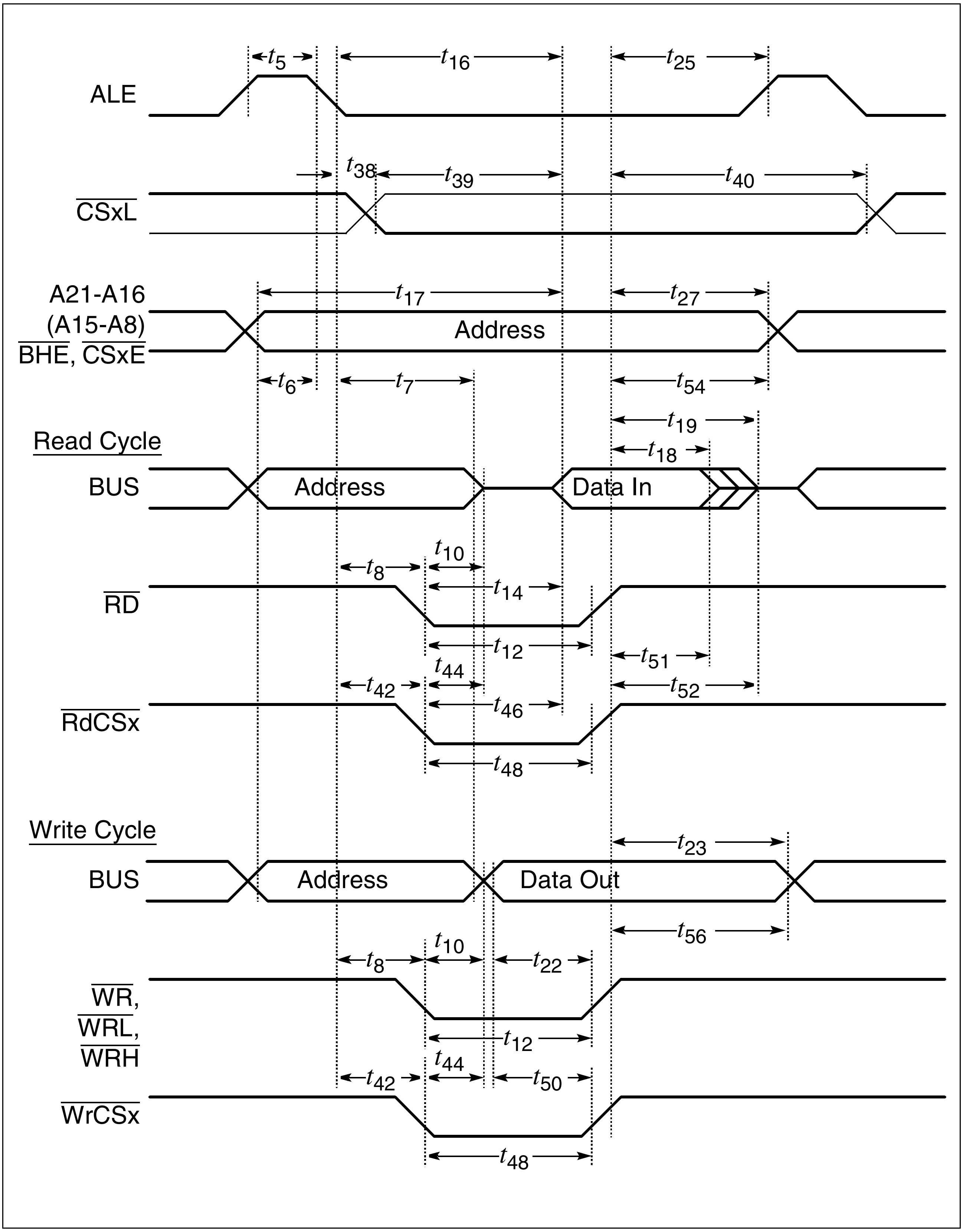


Figure 16 External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE

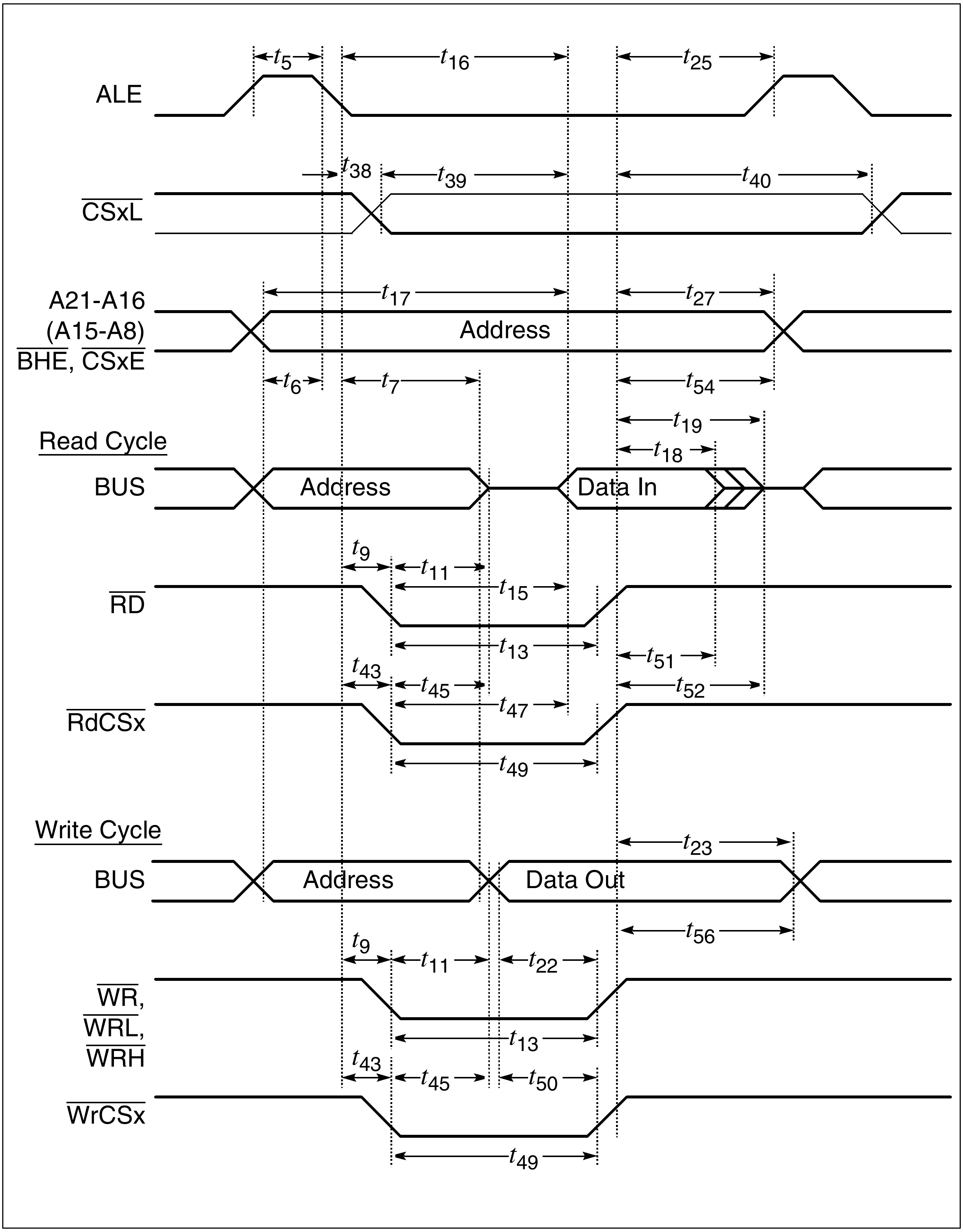


Figure 18 External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Normal ALE

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 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).

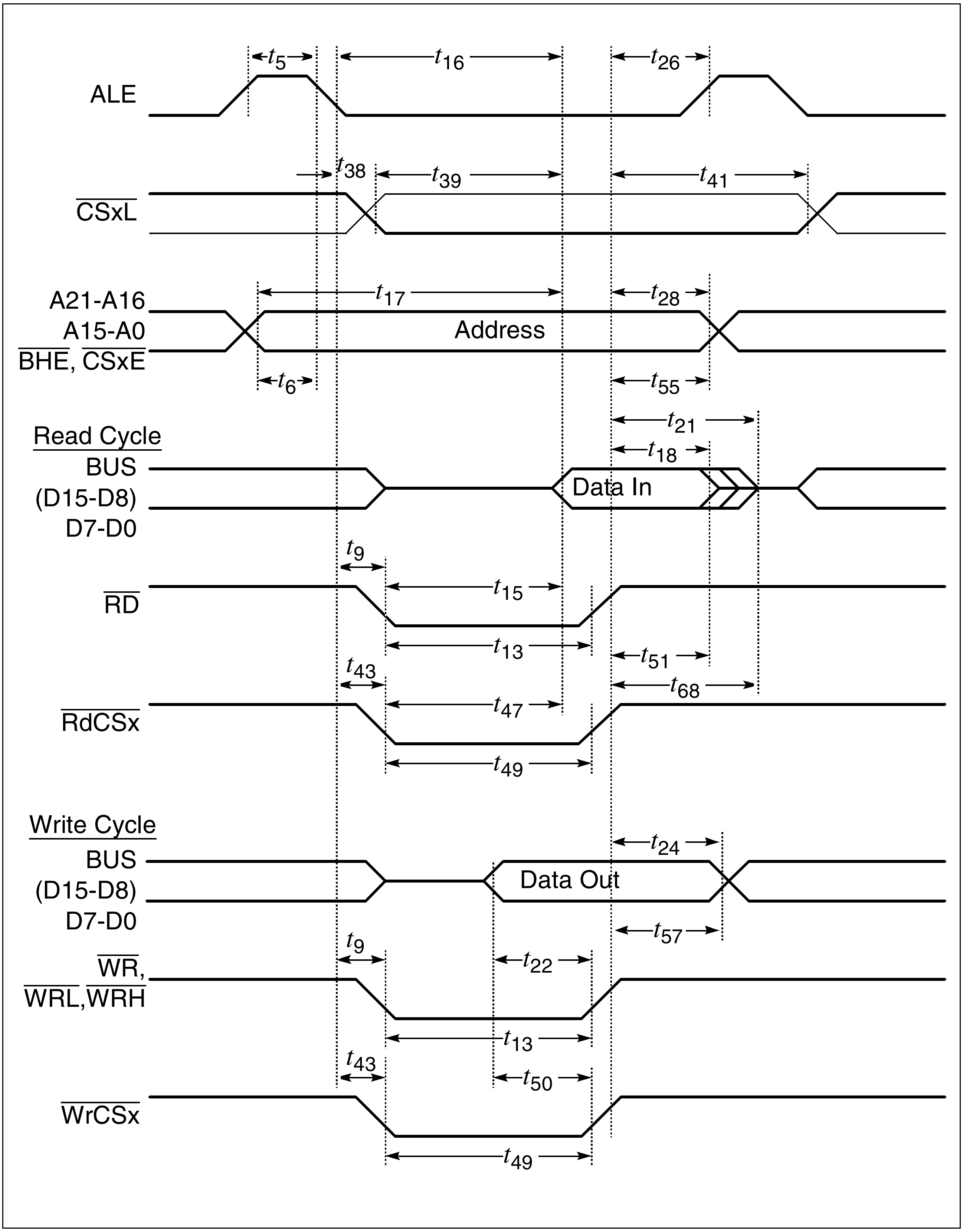


Figure 22 **External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Normal ALE**

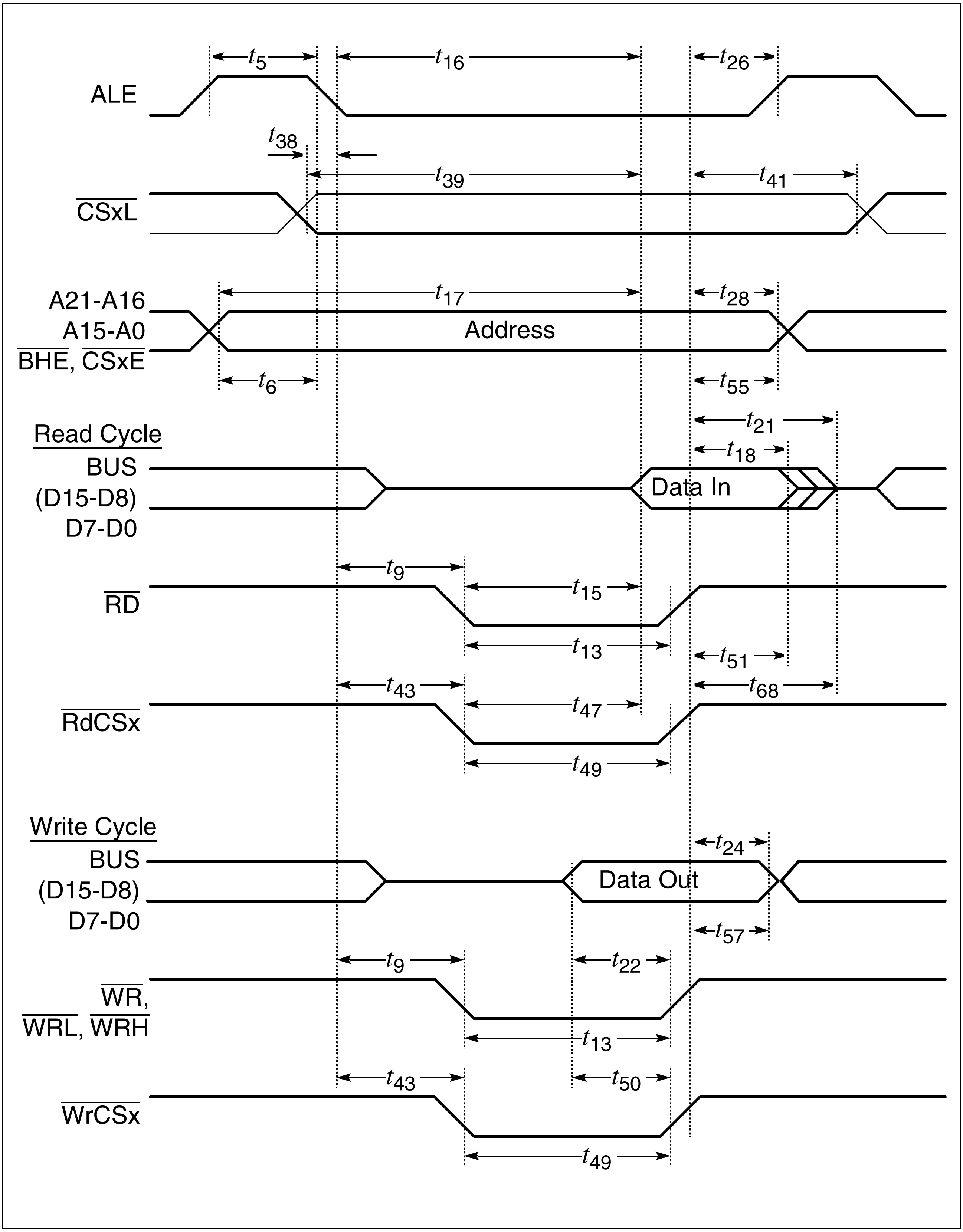


Figure 23 External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Extended ALE

External XRAM Access

If XPER-Share mode is enabled the on-chip XRAM of the C164CI can be accessed (during hold states) by an external master like an asynchronous SRAM.

Table 16 XRAM Access Timing (Operating Conditions apply)

Parameter		Symbol		Limit Values		Unit
				min.	max.	
Address setup time before $\overline{\text{RD}}/\overline{\text{WR}}$ falling edge		t_{40}	SR	4	–	ns
Address hold time after $\overline{\text{RD}}/\overline{\text{WR}}$ rising edge		t_{41}	SR	0	–	ns
Data turn on delay after $\overline{\text{RD}}$ falling edge	Read	t_{42}	CC	2	–	ns
Data output valid delay after address latched	Read	t_{43}	CC	–	37	ns
Data turn off delay after $\overline{\text{RD}}$ rising edge	Read	t_{44}	CC	0	10	ns
Write data setup time before $\overline{\text{WR}}$ rising edge	Write	t_{45}	SR	10	–	ns
Write data hold time after $\overline{\text{WR}}$ rising edge	Write	t_{46}	SR	1	–	ns
$\overline{\text{WR}}$ pulse width	Write	t_{47}	SR	18	–	ns
$\overline{\text{WR}}$ signal recovery time	Write	t_{48}	SR	t_{40}	–	ns

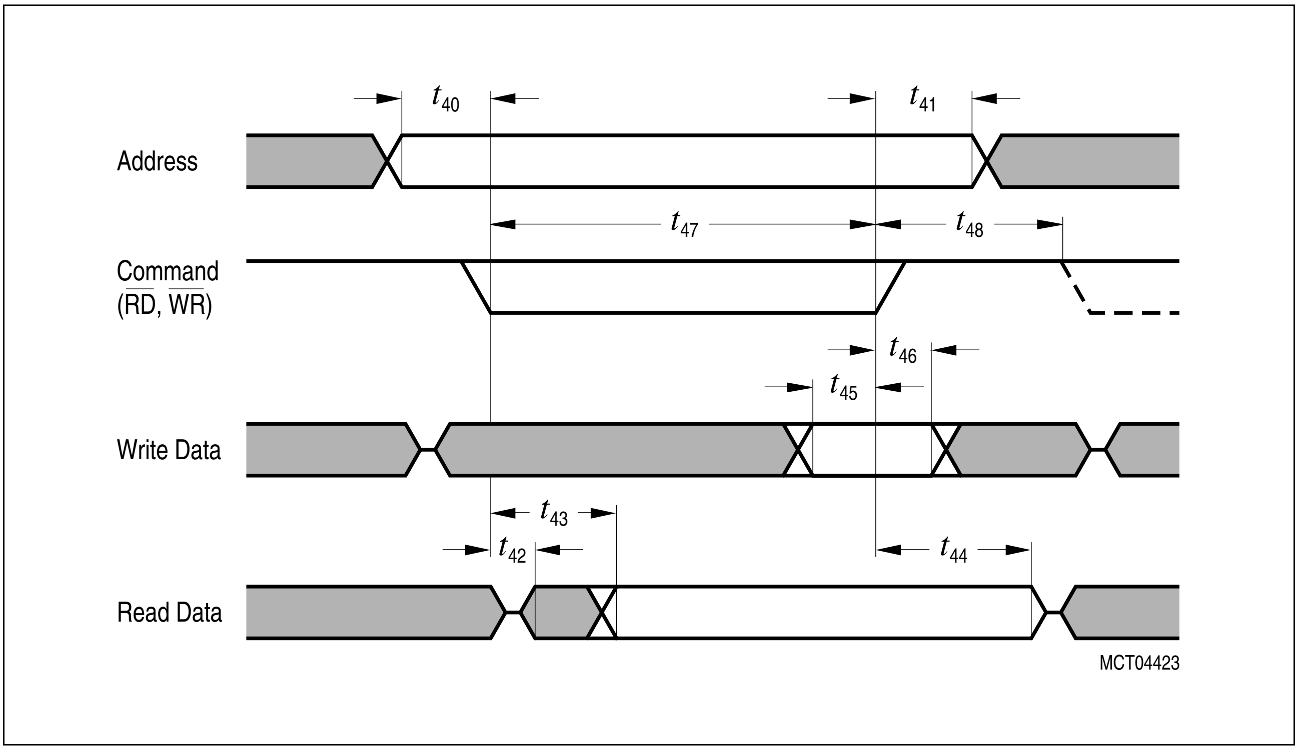


Figure 25 External Access to the XRAM