Infineon Technologies - C164CI8E28MDBFXUMA1 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	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 ~ 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/c164ci8e28mdbfxuma1

Data Sheet, V2.0, May 2001

C164CI/SI
C164CL/SL

16-Bit Single-Chip Microcontroller

Microcontrollers

Never stop thinking.

Table 2 Pin Definitions and Functions

Symbol	Pin No.	Input Outp.	Function		
P5		I	Port 5 is an 8-bit input-only port with Schmitt-Trigger charact. The pins of Port 5 also serve as analog input channels for the A/D converter, or they serve as timer inputs:		
P5.0	76	I	AN0		
P5.1	77	I	AN1		
P5.2	78	I	AN2		
P5.3	79	I	AN3		
P5.4	2	I	AN4,	T2EUD	GPT1 Timer T2 Ext. Up/Down Ctrl. Inp.
P5.5	3	I	AN5,	T4EUD	GPT1 Timer T4 Ext. Up/Down Ctrl. Inp.
P5.6	4	I	AN6,	T2IN	GPT1 Timer T2 Input for Count/Gate/Reload/Capture
P5.7	5	I	AN7,	T4IN	GPT1 Timer T4 Input for Count/Gate/Reload/Capture
P3		IO	Port 3 is a 9-bit bidirectional I/O port. 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. Port 3 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 3 is selectable (TTL or special). The following Port 3 pins also serve for alternate functions:		
P3.4	8	I	T3EUD	GPT1 Timer T3 External Up/Down Control Input	
P3.6	9	I	T3IN	GPT1 Timer T3 Count/Gate Input	
P3.8	10	I/O	MRST	SSC Master-Receive/Slave-Transmit Inp./Outp.	
P3.9	11	I/O	MTSR	SSC Master-Transmit/Slave-Receive Outp./Inp.	
P3.10	12	O	TxD0	ASC0 Clock/Data Output (Async./Sync.)	
P3.11	13	I/O	RxD0	ASC0 Data Input (Async.) or Inp./Outp. (Sync.)	
P3.12	14	O	$\overline{\text{BHE}}$	External Memory High Byte Enable Signal,	
		O	$\overline{\text{WRH}}$	External Memory High Byte Write Strobe	
P3.13	15	I/O	SCLK	SSC Master Clock Output / Slave Clock Input.	
P3.15	16	O	CLKOUT	System Clock Output (= CPU Clock),	
		O	FOUT	Programmable Frequency Output	

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.

Table 2 Pin Definitions and Functions (cont'd)

Symbol	Pin No.	Input Outp.	Function
P8		IO	Port 8 is a 4-bit bidirectional I/O port. 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. Port 8 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 8 is selectable (TTL or special). Port 8 pins provide inputs/outputs for CAPCOM2 and serial interface lines.[1)]
P8.0	72	I/O I	CC16IO CAPCOM2: CC16 Capture Inp./Compare Outp., CAN1_RxD CAN 1 Receive Data Input
P8.1	73	I/O O	CC17IO CAPCOM2: CC17 Capture Inp./Compare Outp., CAN1_TxD CAN 1 Transmit Data Output
P8.2	74	I/O I	CC18IO CAPCOM2: CC18 Capture Inp./Compare Outp., CAN1_RxD CAN 1 Receive Data Input
P8.3	75	I/O O	CC19IO CAPCOM2: CC19 Capture Inp./Compare Outp., CAN1_TxD CAN 1 Transmit Data Output
V_{AREF}	1	–	Reference voltage for the A/D converter.
V_{AGND}	80	–	Reference ground for the A/D converter.
V_{DD}	7, 21, 40, 53, 61	–	Digital Supply Voltage: +5 V during normal operation and idle mode. ≥2.5 V during power down mode.
V_{SS}	6, 20, 41, 56, 60	–	Digital Ground.

1) The CAN interface lines are assigned to ports P4 and P8 under software control. Within the CAN module several assignments can be selected.

Note: The following behavioural differences must be observed when the bidirectional reset is active:

- Bit BDRSTEN in register SYSCON cannot be changed after EINIT and is cleared automatically after a reset.
- The reset indication flags always indicate a long hardware reset.
- The PORT0 configuration is treated as if it were a hardware reset. In particular, the bootstrap loader may be activated when P0L.4 is low.
- Pin $\overline{\text{RSTIN}}$ may only be connected to external reset devices with an open drain output driver.
- A short hardware reset is extended to the duration of the internal reset sequence.

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).

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 – Software Reset – W-dog Timer Overflow	–	 RESET RESET RESET	 $00'0000_H$ $00'0000_H$ $00'0000_H$	 00_H 00_H 00_H	 III III III
Class A Hardware Traps: – Non-Maskable Interrupt – Stack Overflow – Stack Underflow	 NMI STKOF STKUF	 NMITRAP STOTRAP STUTRAP	 $00'0008_H$ $00'0010_H$ $00'0018_H$	 02_H 04_H 06_H	 II II II
Class B Hardware Traps: – Undefined Opcode – Protected Instruction Fault – Illegal Word Operand Access – Illegal Instruction Access – Illegal External Bus Access	 UNDOPC PRTFLT ILLOPA ILLINA ILLBUS	 BTRAP BTRAP BTRAP BTRAP BTRAP	 $00'0028_H$ $00'0028_H$ $00'0028_H$ $00'0028_H$ $00'0028_H$	 $0A_H$ $0A_H$ $0A_H$ $0A_H$ $0A_H$	 I I I I 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

Power Management

The C164CI 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 C164CI into a special operating mode (control via instructions).
  Idle Mode stops the CPU while the peripherals can continue to operate.
  Sleep Mode and Power Down Mode stop all clock signals and all operation (RTC may optionally continue running). Sleep Mode can be terminated by external interrupt signals.
- **Clock Generation Management** controls the distribution and the frequency of internal and external clock signals (control via register SYSCON2).
  Slow Down Mode lets the C164CI 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.
  External circuitry can be controlled via the programmable frequency output FOUT.
- **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 intermittend operation of the C164CI by generating cyclic wakeup signals. This offers full performance to quickly react on action requests while the intermittend sleep phases greatly reduce the average power consumption of the system.

Table 6 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 und 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 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

Power Consumption C164CI (OTP)

(Operating Conditions apply)

Parameter	Symbol	Limit Values		Unit	Test Conditions
		min.	max.		
Power supply current (active) with all peripherals active	I_{DD}	–	10 + 3.5 × f_{CPU}	mA	$\overline{RSTIN}$ = V_{IL} f_{CPU} in [MHz] [1]
Idle mode supply current with all peripherals active	I_{IDX}	–	5 + 1.25 × 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 10**. 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.

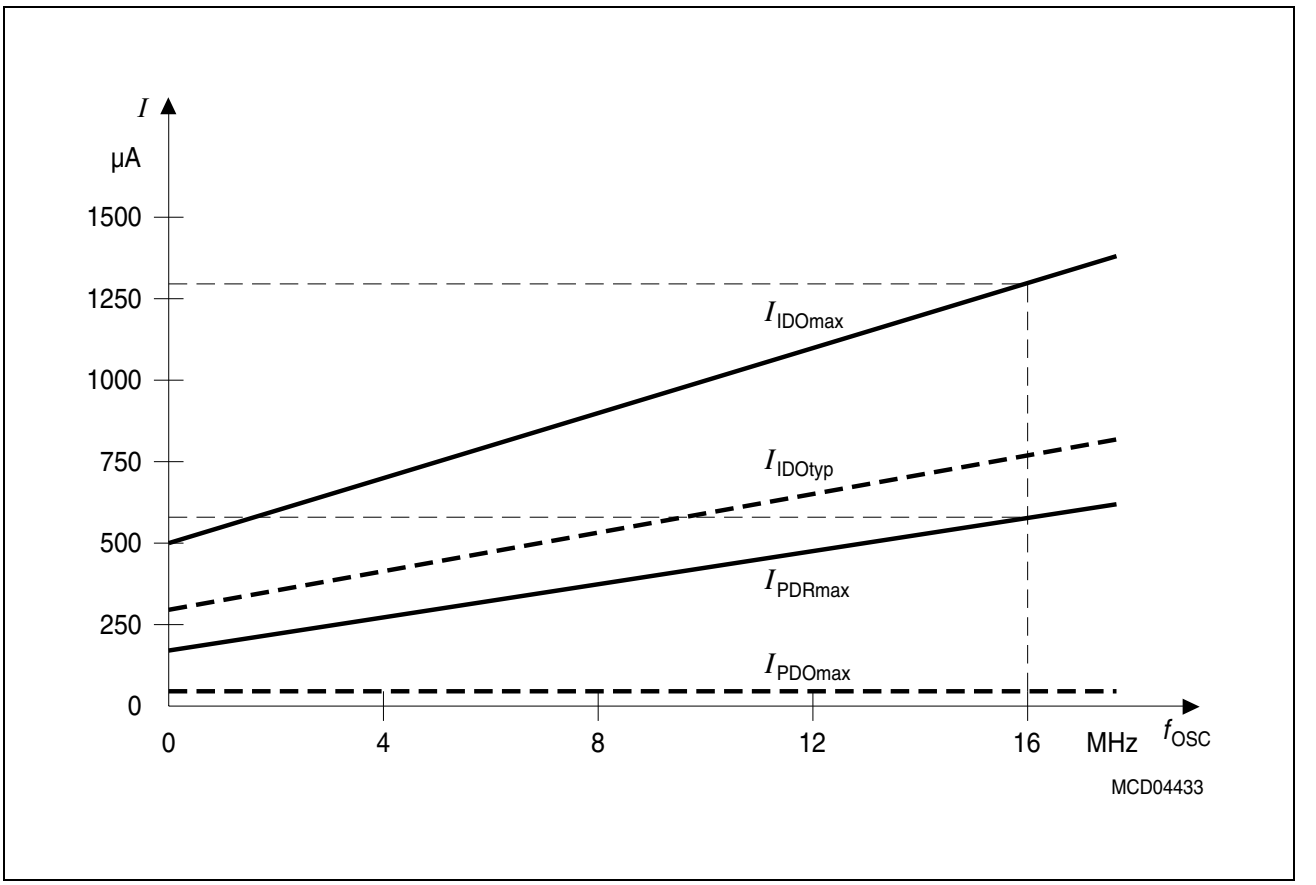

Figure 8 **Idle and Power Down Supply Current as a Function of Oscillator Frequency**

Due to this adaptation to the input clock the frequency of f_{CPU} is constantly adjusted so it is locked to f_{OSC}. The slight variation causes a jitter of f_{CPU} which also effects the duration of individual TCLs.

The timings listed in the AC Characteristics that refer to TCLs therefore must be calculated using the minimum TCL that is possible under the respective circumstances. The actual minimum value for TCL depends on the jitter of the PLL. As the PLL is constantly adjusting its output frequency so it corresponds to the applied input frequency (crystal or oscillator) the relative deviation for periods of more than one TCL is lower than for one single TCL (see formula and **Figure 12**).
For a period of $N \times$ TCL the minimum value is computed using the corresponding deviation D_N:

$$(N \times \text{TCL})_{min} = N \times \text{TCL}_{NOM} - D_N;\ D_N\ [\text{ns}] = \pm(13.3 + N \times 6.3)/f_{CPU}\ [\text{MHz}],$$

where N = number of consecutive TCLs and $1 \le N \le 40$.

So for a period of 3 TCLs @ 25 MHz (i.e. $N = 3$): $D_3 = (13.3 + 3 \times 6.3)/25 = 1.288$ ns, and $(3\text{TCL})_{min} = 3\text{TCL}_{NOM} - 1.288$ ns = 58.7 ns (@ f_{CPU} = 25 MHz).

This is especially important for bus cycles using waitstates and e.g. for the operation of timers, serial interfaces, etc. For all slower operations and longer periods (e.g. pulse train generation or measurement, lower baudrates, etc.) the deviation caused by the PLL jitter is neglectible.

*Note: For all periods longer than 40 TCL the N = 40 value can be used (see **Figure 12**).*

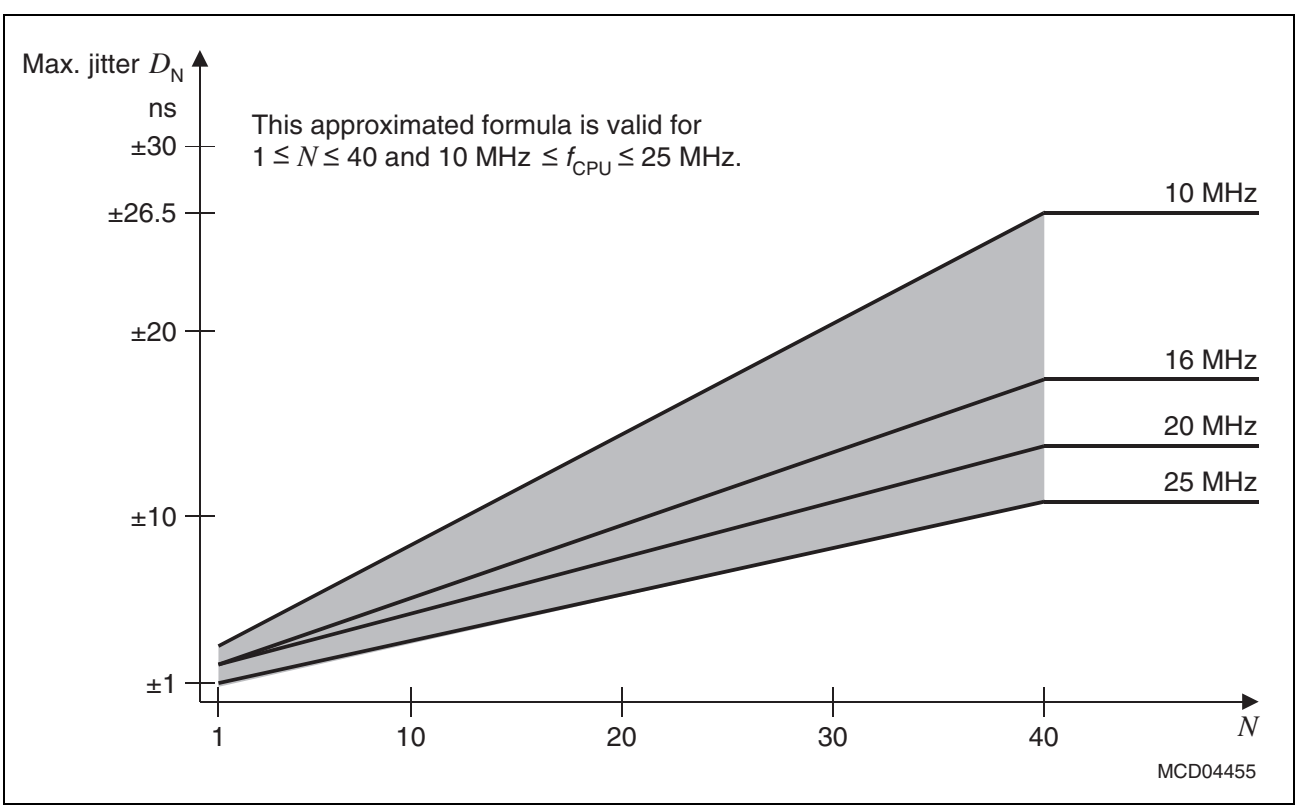

Figure 12 Approximated Maximum Accumulated PLL Jitter

Direct Drive

When direct drive is configured (CLKCFG = 011_B) the on-chip phase locked loop is disabled and the CPU clock is directly driven from the internal oscillator with the input clock signal.
The frequency of f_{CPU} directly follows the frequency of f_{OSC} so the high and low time of f_{CPU} (i.e. the duration of an individual TCL) is defined by the duty cycle of the input clock f_{OSC}.
The timings listed below that refer to TCLs therefore must be calculated using the minimum TCL that is possible under the respective circumstances. This minimum value can be calculated via the following formula:

$$TCL_{min} = 1/f_{OSC} \times DC_{min} \qquad \text{(DC = duty cycle)}$$

For two consecutive TCLs the deviation caused by the duty cycle of f_{OSC} is compensated so the duration of 2TCL is always $1/f_{OSC}$. The minimum value TCL_{min} therefore has to be used only once for timings that require an odd number of TCLs (1, 3, …). Timings that require an even number of TCLs (2, 4, …) may use the formula $2TCL = 1/f_{OSC}$.

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. The following table 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.

AC Characteristics

Multiplexed Bus

(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 $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8	CC	10 + t_A	–	TCL - 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9	CC	-10 + t_A	–	-10 + t_A	–	ns
Address float after $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_{10}	CC	–	6	–	6	ns
Address float after $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_{11}	CC	–	26	–	TCL + 6	ns
$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	t_{12}	CC	30 + t_C	–	2TCL - 10 + t_C	–	ns

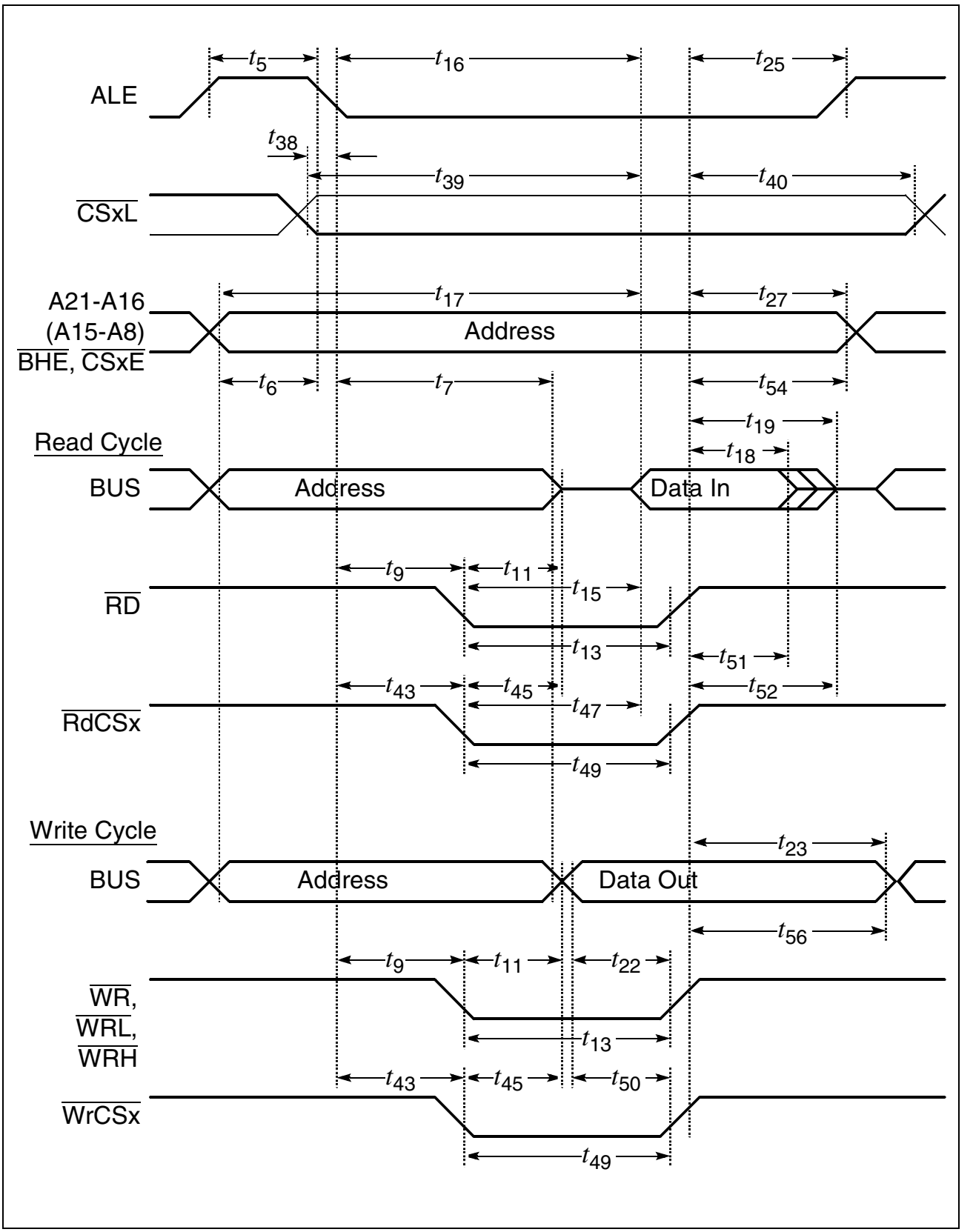

**Figure 19 External Memory Cycle:
Multiplexed Bus, No Read/Write Delay, Extended ALE**

AC Characteristics

Demultiplexed Bus

(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.	
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
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8	CC	10 + t_A	–	TCL - 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9	CC	-10 + t_A	–	-10 + t_A	–	ns
$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	t_{12}	CC	30 + t_C	–	2TCL - 10 + t_C	–	ns
$\overline{RD}$, $\overline{WR}$ 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 $\overline{RD}$ rising edge (with RW-delay[1)])	t_{20}	SR	–	26 + $2t_A$ + t_F[1)]	–	2TCL - 14 + $22t_A$ + t_F[1)]	ns
Data float after $\overline{RD}$ rising edge (no RW-delay[1)])	t_{21}	SR	–	10 + $2t_A$ + t_F[1)]	–	TCL - 10 + $22t_A$ + 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 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

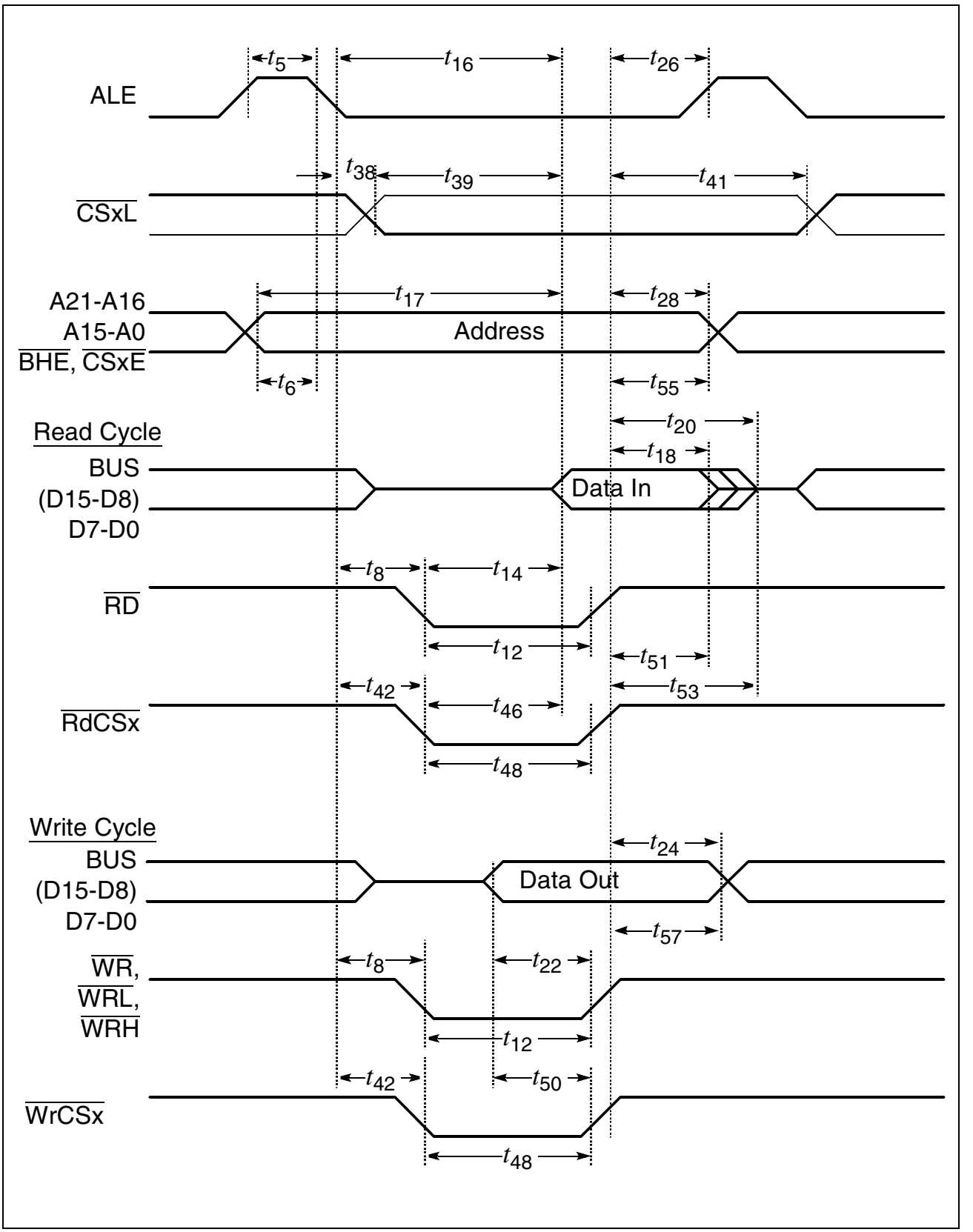

Figure 20 External Memory Cycle:
Demultiplexed Bus, With Read/Write Delay, Normal ALE

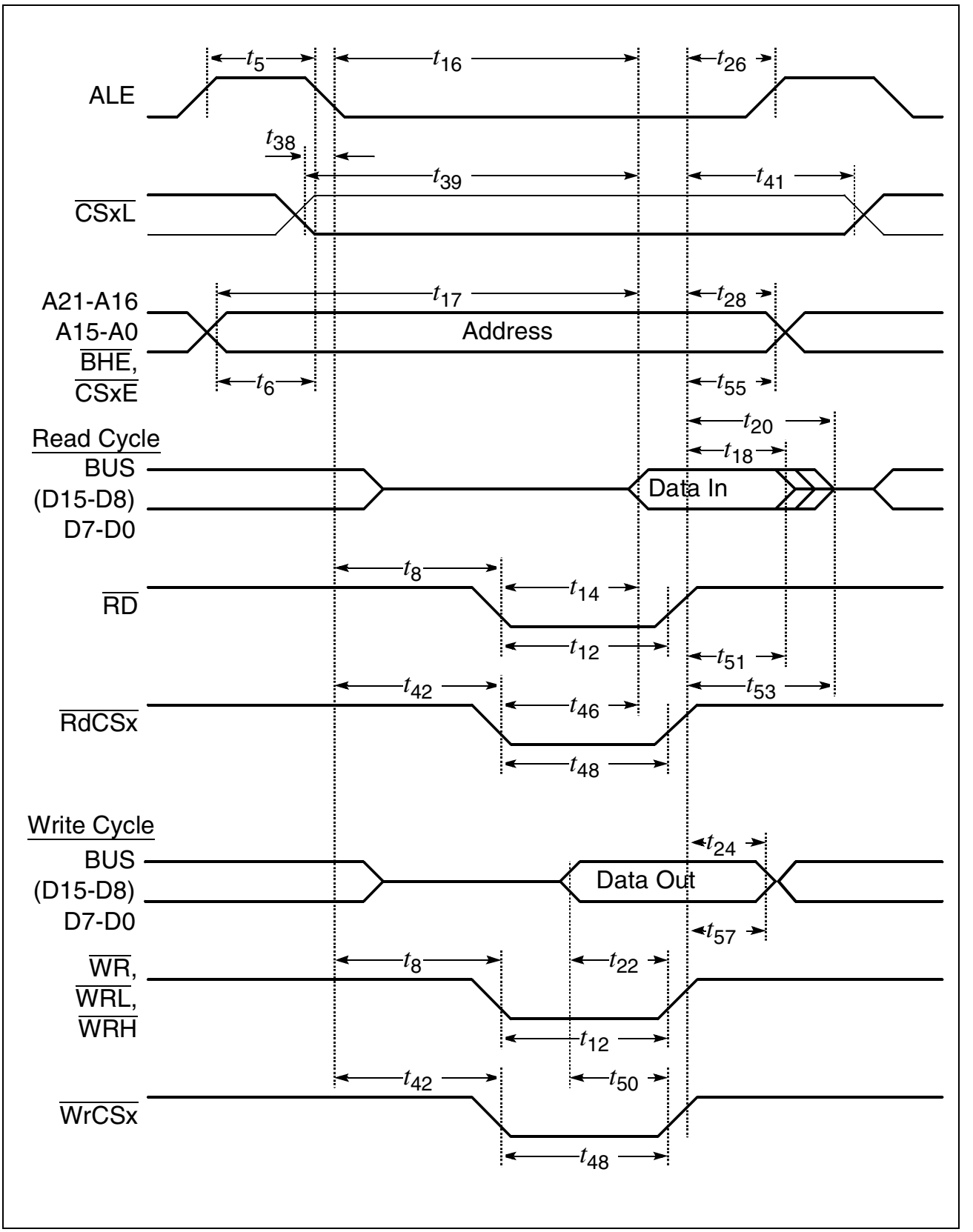

Figure 21 External Memory Cycle:
Demultiplexed Bus, With Read/Write Delay, Extended ALE