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	Discontinued at Digi-Key
Core Processor	C166
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, I²C, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	93
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	10K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	128-LQFP
Supplier Device Package	PG-TQFP-128-2
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/sak-c161cs-lf-ca

Ordering Information

The ordering code for Infineon microcontrollers provides an exact reference to the required product. This ordering code identifies:

- the derivative itself, i.e. its function set, the temperature range, and the supply voltage
- the package and the type of delivery.

For the available ordering codes for the C161CS/JC/JI please refer to the **“Product Catalog Microcontrollers”**, which summarizes all available microcontroller variants.

Note: The ordering codes for Mask-ROM versions are defined for each product after verification of the respective ROM code.

Introduction

The C161CS/JC/JI derivatives are high performance derivatives of the Infineon C166 Family of full featured single-chip CMOS microcontrollers. They combine high CPU performance (up to 12.5 million instructions per second) with high peripheral functionality and enhanced IO-capabilities. They also provide clock generation via PLL and various on-chip memory modules such as program ROM, internal RAM, and extension RAM.

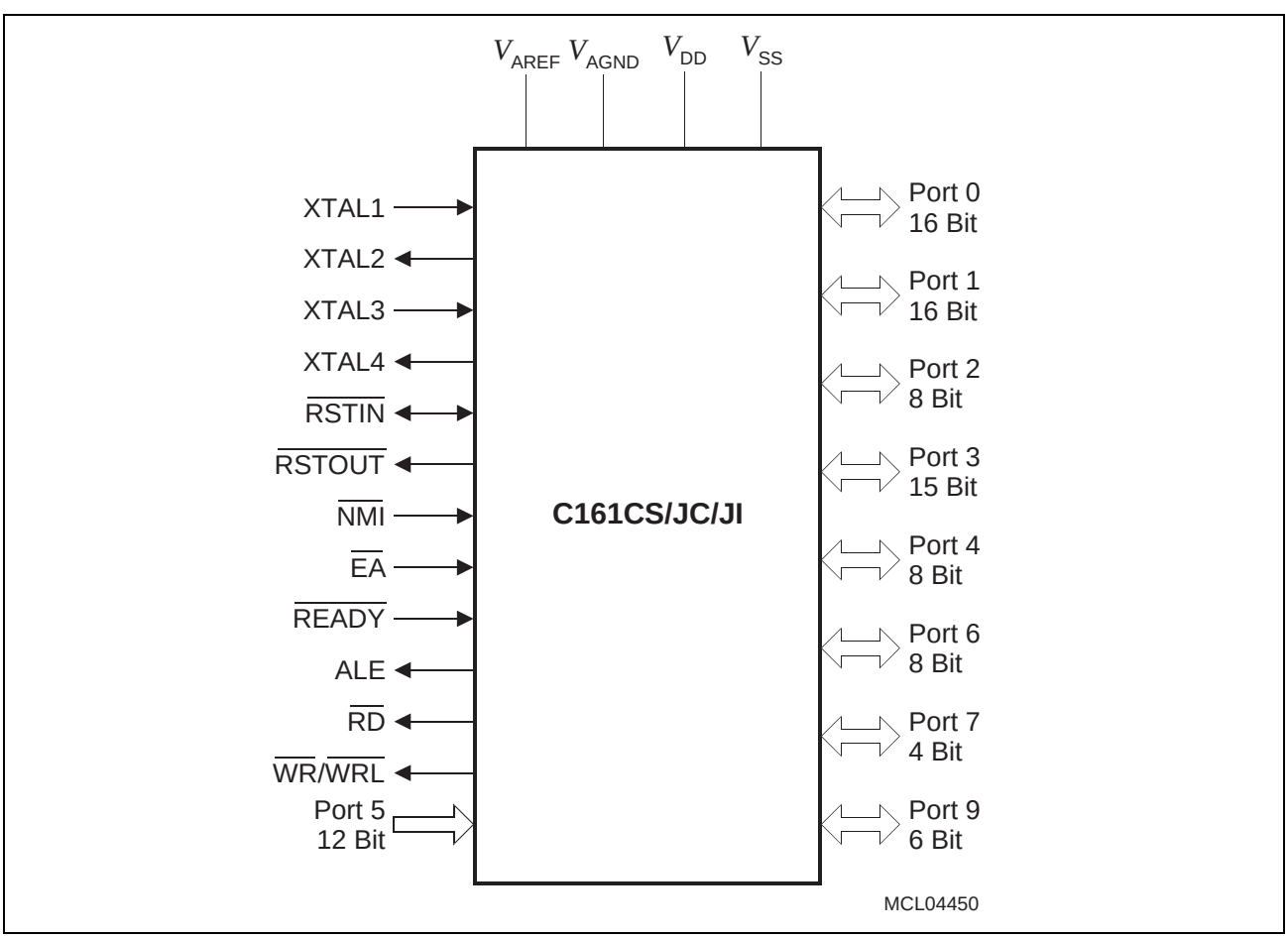

Figure 1 Logic Symbol

Table 2 Pin Definitions and Functions (cont'd)

Symbol	Pin No.	Input Outp.	Function
P7		IO	Port 7 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 7 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 7 is selectable (TTL or special). Port 7 pins provide inputs/outputs for CAPCOM2 and serial interface lines.[1)]
P7.4	13	I/O	CC28IO CAPCOM2: CC28 Capture Inp./Compare Outp.,
		I	CAN1_RxD CAN 1 Receive Data Input, (**C161CS/JC**)
		I	CAN2_RxD CAN 2 Receive Data Input, (**C161CS**)
		O	SDL_TxD SDLM Transmit Data Output (**C161JC/JI**)
P7.5	14	I/O	CC29IO CAPCOM2: CC29 Capture Inp./Compare Outp.,
		O	CAN1_TxD CAN 1 Transmit Data Output, (**C161CS/JC**)
		O	CAN2_TxD CAN 2 Transmit Data Output, (**C161CS**)
		I	SDL_RxD SDLM Receive Data Input (**C161JC/JI**)
P7.6	15	I/O	CC30IO CAPCOM2: CC30 Capture Inp./Compare Outp.,
		I	CAN1_RxD CAN 1 Receive Data Input, (**C161CS/JC**)
		I	CAN2_RxD CAN 2 Receive Data Input, (**C161CS**)
		O	SDL_TxD SDLM Transmit Data Output (**C161JC/JI**)
P7.7	16	I/O	CC31IO CAPCOM2: CC31 Capture Inp./Compare Outp.,
		O	CAN1_TxD CAN 1 Transmit Data Output, (**C161CS/JC**)
		O	CAN2_TxD CAN 2 Transmit Data Output, (**C161CS**)
		I	SDL_RxD SDLM Receive Data Input (**C161JC/JI**)
P9		IO	Port 9 is a 6-bit bidirectional open drain I/O port (provide external pullup resistors if required). 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. The following Port 9 pins also serve for alternate functions:
P9.0	19	I/O	SDA0 IIC Bus Data Line 0
P9.1	20	I/O	SCL0 IIC Bus Clock Line 0
P9.2	21	I/O	SDA1 IIC Bus Data Line 1
P9.3	22	I/O	SCL1 IIC Bus Clock Line 1
P9.4	23	I/O	SDA2 IIC Bus Data Line 2
P9.5	24	–	–
			*Note: Port 9 pins can only tolerate positive overload currents (see **Table 9**).*

Table 2 Pin Definitions and Functions (cont'd)

Symbol	Pin No.	Input Outp.	Function	
P3		IO	Port 3 is a 15-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:	
P3.0	53	I	T0IN	CAPCOM1 Timer T0 Count Input,
		O	TxD1	ASC1 Clock/Data Output (Async./Sync)
P3.1	54	O	T6OUT	GPT2 Timer T6 Toggle Latch Output,
		I/O	RxD1	ASC1 Data Input (Async.) or Inp./Output (Sync.)
P3.2	55	I	CAPIN	GPT2 Register CAPREL Capture Input
P3.3	56	O	T3OUT	GPT1 Timer T3 Toggle Latch Output
P3.4	57	I	T3EUD	GPT1 Timer T3 External Up/Down Control Input
P3.5	58	I	T4IN	GPT1 Timer T4 Count/Gate/Reload/Capture Inp
P3.6	59	I	T3IN	GPT1 Timer T3 Count/Gate Input
P3.7	60	I	T2IN	GPT1 Timer T2 Count/Gate/Reload/Capture Inp
P3.8	61	I/O	MRST	SSC Master-Receive/Slave-Transmit Inp./Outp.
P3.9	62	I/O	MTSR	SSC Master-Transmit/Slave-Receive Outp./Inp.
P3.10	63	O	TxD0	ASC0 Clock/Data Output (Async./Sync.)
P3.11	64	I/O	RxD0	ASC0 Data Input (Async.) or Inp./Outp. (Sync.)
P3.12	65	O	$\overline{\text{BHE}}$	External Memory High Byte Enable Signal,
		O	$\overline{\text{WRH}}$	External Memory High Byte Write Strobe
P3.13	66	I/O	SCLK	SSC Master Clock Output / Slave Clock Input.
P3.15	67	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
P4		IO	Port 4 is an 8-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. The Port 4 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 4 is selectable (TTL or special). Port 4 can be used to output the segment address lines and for serial interface lines:[1)]
P4.0	70	O	A16 Least Significant Segment Address Line
P4.1	71	O	A17 Segment Address Line
P4.2	72	O	A18 Segment Address Line
P4.3	73	O	A19 Segment Address Line
P4.4	74	O	A20 Segment Address Line,
		I	CAN2_RxD CAN 2 Receive Data Input, (**C161CS**)
		I	SDL_RxD SDLM Receive Data Input (**C161JC/JI**)
P4.5	75	O	A21 Segment Address Line,
		I	CAN1_RxD CAN 1 Receive Data Input, (**C161CS/JC**)
P4.6	76	O	A22 Segment Address Line,
		O	CAN1_TxD CAN 1 Transmit Data Output, (**C161CS/JC**)
		O	CAN2_TxD CAN 2 Transmit Data Output, (**C161CS**)
		I	SDL_RxD SDLM Receive Data Input (**C161JC/JI**)
P4.7	77	O	A23 Most Significant Segment Address Line,
		I	CAN1_RxD CAN 1 Receive Data Input, (**C161CS/JC**)
		O	CAN2_TxD CAN 2 Transmit Data Output, (**C161CS**)
		I	CAN2_RxD CAN 2 Receive Data Input, (**C161CS**)
		O	SDL_TxD SDLM Transmit Data Output (**C161JC/JI**)
$\overline{RD}$	80	O	External Memory Read Strobe. $\overline{RD}$ is activated for every external instruction or data read access.
$\overline{WR}$/ $\overline{WRL}$	81	O	External Memory Write Strobe. In $\overline{WR}$-mode this pin is activated for every external data write access. In $\overline{WRL}$-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.

When a match occurs between the timer value and the value in a capture/compare register, specific actions will be taken based on the selected compare mode.

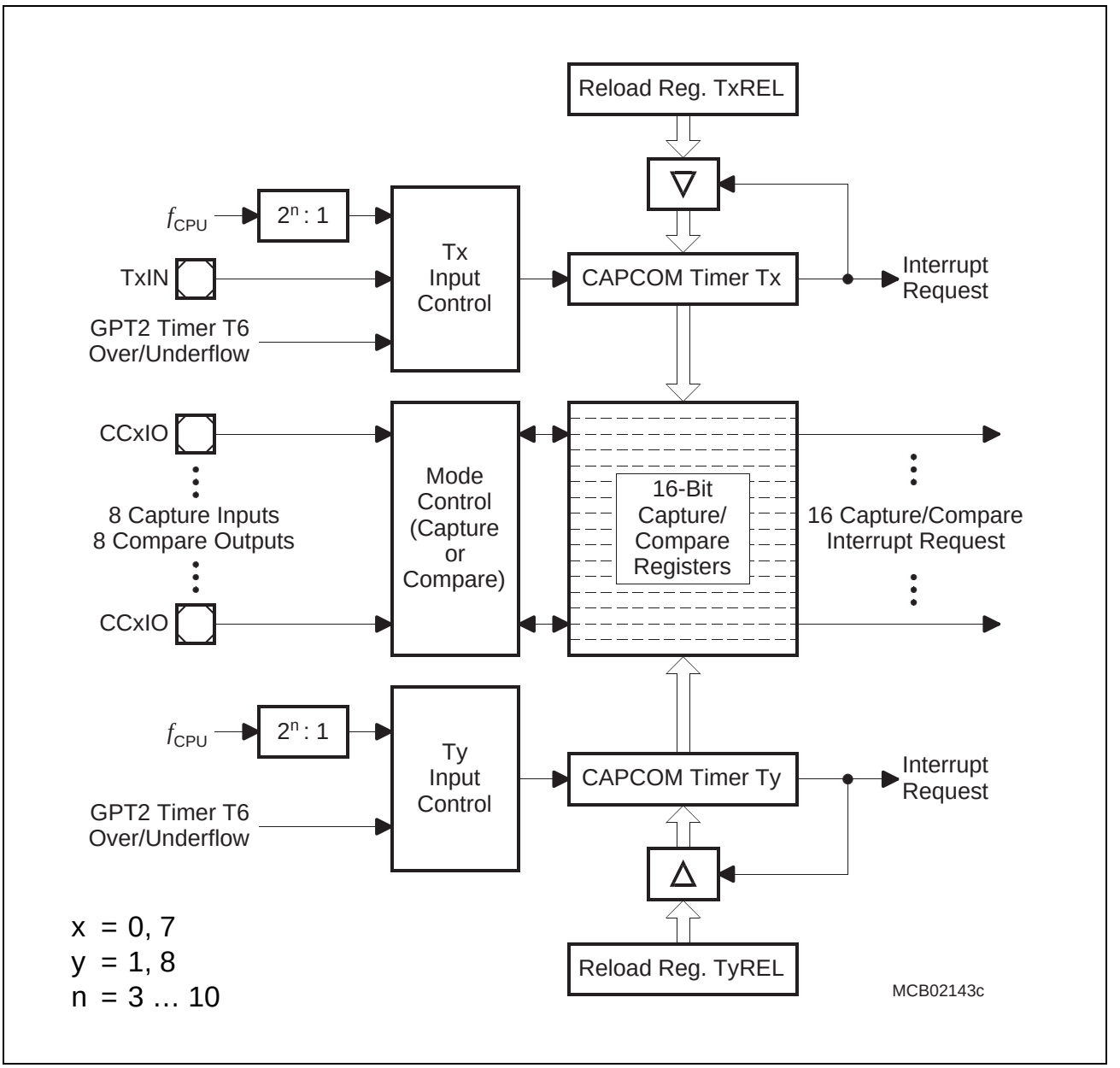

Figure 5 CAPCOM Unit Block Diagram

Instruction Set Summary

Table 6 lists the instructions of the C161CS/JC/JI in a condensed way.
The various addressing modes that can be used with a specific instruction, the operation of the instructions, parameters for conditional execution of instructions, and the opcodes for each instruction can be found in the **“C166 Family Instruction Set Manual”**.

This document also provides a detailled description of each instruction.

Table 6 Instruction Set Summary

Mnemonic	Description	Bytes
ADD(B)	Add word (byte) operands	2 / 4
ADDC(B)	Add word (byte) operands with Carry	2 / 4
SUB(B)	Subtract word (byte) operands	2 / 4
SUBC(B)	Subtract word (byte) operands with Carry	2 / 4
MUL(U)	(Un)Signed multiply direct GPR by direct GPR (16-16-bit)	2
DIV(U)	(Un)Signed divide register MDL by direct GPR (16-/16-bit)	2
DIVL(U)	(Un)Signed long divide reg. MD by direct GPR (32-/16-bit)	2
CPL(B)	Complement direct word (byte) GPR	2
NEG(B)	Negate direct word (byte) GPR	2
AND(B)	Bitwise AND, (word/byte operands)	2 / 4
OR(B)	Bitwise OR, (word/byte operands)	2 / 4
XOR(B)	Bitwise XOR, (word/byte operands)	2 / 4
BCLR	Clear direct bit	2
BSET	Set direct bit	2
BMOV(N)	Move (negated) direct bit to direct bit	4
BAND, BOR, BXOR	AND/OR/XOR direct bit with direct bit	4
BCMP	Compare direct bit to direct bit	4
BFLDH/L	Bitwise modify masked high/low byte of bit-addressable direct word memory with immediate data	4
CMP(B)	Compare word (byte) operands	2 / 4
CMPD1/2	Compare word data to GPR and decrement GPR by 1/2	2 / 4
CMPI1/2	Compare word data to GPR and increment GPR by 1/2	2 / 4
PRIOR	Determine number of shift cycles to normalize direct word GPR and store result in direct word GPR	2
SHL / SHR	Shift left/right direct word GPR	2
ROL / ROR	Rotate left/right direct word GPR	2
ASHR	Arithmetic (sign bit) shift right direct word GPR	2

Table 7 C161CS/JC/JI Registers, Ordered by Name (cont'd)

Name	Physical Address	8-Bit Addr.	Description	Reset Value
C1PCIR	$EF02_H$ **X**	---	CAN1 Port Control / Interrupt Register	$XXXX_H$
C1LARn	$EFn4_H$ **X**	---	CAN1 Lower Arbitration Reg. (msg. **n**)	$UUUU_H$
C1LGML	$EF0A_H$ **X**	---	CAN1 Lower Global Mask Long	$UUUU_H$
C1LMLM	$EF0E_H$ **X**	---	CAN1 Lower Mask of Last Message	$UUUU_H$
C1MCFGn	$EFn6_H$ **X**	---	CAN1 Message Config. Reg. (msg. **n**)	UU_H
C1MCRn	$EFn0_H$ **X**	---	CAN1 Message Control Reg. (msg. **n**)	$UUUU_H$
C1UARn	$EFn2_H$ **X**	---	CAN1 Upper Arbitration Reg. (msg. **n**)	$UUUU_H$
C1UGML	$EF08_H$ **X**	---	CAN1 Upper Global Mask Long	$UUUU_H$
C1UMLM	$EF0C_H$ **X**	---	CAN1 Upper Mask of Last Message	$UUUU_H$
C2BTR	$EE04_H$ **X**	---	CAN2 Bit Timing Register	$UUUU_H$
C2CSR	$EE00_H$ **X**	---	CAN2 Control / Status Register	$XX01_H$
C2GMS	$EE06_H$ **X**	---	CAN2 Global Mask Short	$UFUU_H$
C2PCIR	$EE02_H$ **X**	---	CAN2 Port Control / Interrupt Register	$XXXX_H$
C2LARn	$EEn4_H$ **X**	---	CAN2 Lower Arbitration Reg. (msg. **n**)	$UUUU_H$
C2LGML	$EE0A_H$ **X**	---	CAN2 Lower Global Mask Long	$UUUU_H$
C2LMLM	$EE0E_H$ **X**	---	CAN2 Lower Mask of Last Message	$UUUU_H$
C2MCFGn	$EEn6_H$ **X**	---	CAN2 Message Config. Reg. (msg. **n**)	UU_H
C2MCRn	$EEn0_H$ **X**	---	CAN2 Message Control Reg. (msg. **n**)	$UUUU_H$
C2UARn	$EEn2_H$ **X**	---	CAN2 Upper Arbitration Reg. (msg. **n**)	$UUUU_H$
C2UGML	$EE08_H$ **X**	---	CAN2 Upper Global Mask Long	$UUUU_H$
C2UMLM	$EE0C_H$ **X**	---	CAN2 Upper Mask of Last Message	$UUUU_H$
CAPREL	$FE4A_H$	25_H	GPT2 Capture/Reload Register	0000_H
CC0	$FE80_H$	40_H	CAPCOM Register 0	0000_H
CC0IC **b**	$FF78_H$	BC_H	CAPCOM Register 0 Interrupt Ctrl. Reg.	0000_H
CC1	$FE82_H$	41_H	CAPCOM Register 1	0000_H
CC10	$FE94_H$	$4A_H$	CAPCOM Register 10	0000_H
CC10IC **b**	$FF8C_H$	$C6_H$	CAPCOM Reg. 10 Interrupt Ctrl. Reg.	0000_H
CC11	$FE96_H$	$4B_H$	CAPCOM Register 11	0000_H
CC11IC **b**	$FF8E_H$	$C7_H$	CAPCOM Reg. 11 Interrupt Ctrl. Reg.	0000_H
CC12	$FE98_H$	$4C_H$	CAPCOM Register 12	0000_H
CC12IC **b**	$FF90_H$	$C8_H$	CAPCOM Reg. 12 Interrupt Ctrl. Reg.	0000_H
CC13	$FE9A_H$	$4D_H$	CAPCOM Register 13	0000_H

Table 7 C161CS/JC/JI Registers, Ordered by Name (cont'd)

Name		Physical Address		8-Bit Addr.	Description	Reset Value
CRIC	**b**	$FF6A_H$		$B5_H$	GPT2 CAPREL Interrupt Ctrl. Reg.	0000_H
CSP		$FE08_H$		04_H	CPU Code Segment Pointer Register (8 bits, not directly writeable)	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
DP7	**b**	$FFD2_H$		$E9_H$	Port 7 Direction Control Register	00_H
DP9	**b**	$FFDA_H$		ED_H	Port 9 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
ERRSTAT		$EB22_H$	**X**	---	SDLM Error Status Register	0000_H
EXICON	**b**	$F1C0_H$	**E**	$E0_H$	External Interrupt Control Register	0000_H
EXISEL	**b**	$F1DA_H$	**E**	ED_H	External Interrupt Source Select Register	0000_H
FLAGRST		$EB28_H$	**X**	---	SDLM Flag Reset Register	0000_H
FOCON	**b**	$FFAA_H$		$D5_H$	Frequency Output Control Register	0000_H
GLOBCON		$EB10_H$	**X**	---	SDLM Global Control Register	0000_H
ICADR		$ED06_H$	**X**	---	IIC Address Register	$0XXX_H$
ICCFG		$ED00_H$	**X**	---	IIC Configuration Register	$XX00_H$
ICCON		$ED02_H$	**X**	---	IIC Control Register	0000_H
ICRTB		$ED08_H$	**X**	---	IIC Receive/Transmit Buffer	XX_H
ICST		$ED04_H$	**X**	---	IIC Status Register	0000_H
IDCHIP		$F07C_H$	**E**	$3E_H$	Identifier	$1XXX_H$
IDMANUF		$F07E_H$	**E**	$3F_H$	Identifier	1820_H
IDMEM		$F07A_H$	**E**	$3D_H$	Identifier	$X040_H$

Table 7 C161CS/JC/JI Registers, Ordered by Name (cont'd)

Name	Physical Address	8-Bit Addr.	Description	Reset Value
RXD18	$EB58_H$ **X**	---	SDLM Receive Data Register 18 (bus)	0000_H
S0BG	$FEB4_H$	$5A_H$	Serial Channel 0 Baud Rate Generator Reload Register	0000_H
S0CON **b**	$FFB0_H$	$D8_H$	Serial Channel 0 Control Register	0000_H
S0EIC **b**	$FF70_H$	$B8_H$	Serial Channel 0 Error Interrupt Ctrl. Reg.	0000_H
S0RBUF	$FEB2_H$	59_H	Serial Channel 0 Receive Buffer Register (read only)	$XXXX_H$
S0RIC **b**	$FF6E_H$	$B7_H$	Serial Channel 0 Receive Interrupt Control Register	0000_H
S0TBIC **b**	$F19C_H$ **E**	CE_H	Serial Channel 0 Transmit Buffer Interrupt Control Register	0000_H
S0TBUF	$FEB0_H$	58_H	Serial Channel 0 Transmit Buffer Register	0000_H
S0TIC **b**	$FF6C_H$	$B6_H$	Serial Channel 0 Transmit Interrupt Control Register	0000_H
S1BG	$EDA4_H$ **X**	---	Serial Channel 1 Baud Rate Generator Reload Register	0000_H
S1CON	$EDA6_H$ **X**	---	Serial Channel 1 Control Register	0000_H
S1RBUF	$EDA2_H$ **X**	---	Serial Channel 1 Receive Buffer Register (read only)	$XXXX_H$
S1TBUF	$EDA0_H$ **X**	---	Serial Channel 1 Transmit Buffer Register	0000_H
SOFPTR	$EB60_H$ **X**	---	SDLM Start-of-Frame Pointer Register	0000_H
SP	$FE12_H$	09_H	CPU System Stack Pointer Register	$FC00_H$
SSCBR	$F0B4_H$ **E**	$5A_H$	SSC Baudrate Register	0000_H
SSCCON **b**	$FFB2_H$	$D9_H$	SSC Control Register	0000_H
SSCEIC **b**	$FF76_H$	BB_H	SSC Error Interrupt Control Register	0000_H
SSCRB	$F0B2_H$ **E**	59_H	SSC Receive Buffer (read only)	$XXXX_H$
SSCRIC **b**	$FF74_H$	BA_H	SSC Receive Interrupt Control Register	0000_H
SSCTB	$F0B0_H$ **E**	58_H	SSC Transmit Buffer (write only)	0000_H
SSCTIC **b**	$FF72_H$	$B9_H$	SSC Transmit Interrupt Control Register	0000_H
STKOV	$FE14_H$	$0A_H$	CPU Stack Overflow Pointer Register	$FA00_H$
STKUN	$FE16_H$	$0B_H$	CPU Stack Underflow Pointer Register	$FC00_H$

Table 7 C161CS/JC/JI Registers, Ordered by Name (cont'd)

Name	Physical Address	8-Bit Addr.	Description	Reset Value
SYSCON b	$FF12_H$	89_H	CPU System Configuration Register	[1]$0XX0_H$
SYSCON1 b	$F1DC_H$ **E**	EE_H	CPU System Configuration Register 1	0000_H
SYSCON2 b	$F1D0_H$ **E**	$E8_H$	CPU System Configuration Register 2	0000_H
SYSCON3 b	$F1D4_H$ **E**	EA_H	CPU System Configuration Register 3	$0X00_H$
T0	$FE50_H$	28_H	CAPCOM Timer 0 Register	0000_H
T01CON b	$FF50_H$	$A8_H$	CAPCOM Timer 0 and Timer 1 Ctrl. Reg.	0000_H
T0IC b	$FF9C_H$	CE_H	CAPCOM Timer 0 Interrupt Ctrl. Reg.	0000_H
T0REL	$FE54_H$	$2A_H$	CAPCOM Timer 0 Reload Register	0000_H
T1	$FE52_H$	29_H	CAPCOM Timer 1 Register	0000_H
T14	$F0D2_H$ **E**	69_H	RTC Timer 14 Register	no
T14REL	$F0D0_H$ **E**	68_H	RTC Timer 14 Reload Register	no
T1IC b	$FF9E_H$	CF_H	CAPCOM Timer 1 Interrupt Ctrl. Reg.	0000_H
T1REL	$FE56_H$	$2B_H$	CAPCOM Timer 1 Reload Register	0000_H
T2	$FE40_H$	20_H	GPT1 Timer 2 Register	0000_H
T2CON b	$FF40_H$	$A0_H$	GPT1 Timer 2 Control Register	0000_H
T2IC b	$FF60_H$	$B0_H$	GPT1 Timer 2 Interrupt Control Register	0000_H
T3	$FE42_H$	21_H	GPT1 Timer 3 Register	0000_H
T3CON b	$FF42_H$	$A1_H$	GPT1 Timer 3 Control Register	0000_H
T3IC b	$FF62_H$	$B1_H$	GPT1 Timer 3 Interrupt Control Register	0000_H
T4	$FE44_H$	22_H	GPT1 Timer 4 Register	0000_H
T4CON b	$FF44_H$	$A2_H$	GPT1 Timer 4 Control Register	0000_H
T4IC b	$FF64_H$	$B2_H$	GPT1 Timer 4 Interrupt Control Register	0000_H
T5	$FE46_H$	23_H	GPT2 Timer 5 Register	0000_H
T5CON b	$FF46_H$	$A3_H$	GPT2 Timer 5 Control Register	0000_H
T5IC b	$FF66_H$	$B3_H$	GPT2 Timer 5 Interrupt Control Register	0000_H
T6	$FE48_H$	24_H	GPT2 Timer 6 Register	0000_H
T6CON b	$FF48_H$	$A4_H$	GPT2 Timer 6 Control Register	0000_H
T6IC b	$FF68_H$	$B4_H$	GPT2 Timer 6 Interrupt Control Register	0000_H
T7	$F050_H$ **E**	28_H	CAPCOM Timer 7 Register	0000_H
T78CON b	$FF20_H$	90_H	CAPCOM Timer 7 and 8 Ctrl. Reg.	0000_H
T7IC b	$F17A_H$ **E**	BD_H	CAPCOM Timer 7 Interrupt Ctrl. Reg.	0000_H
T7REL	$F054_H$ **E**	$2A_H$	CAPCOM Timer 7 Reload Register	0000_H

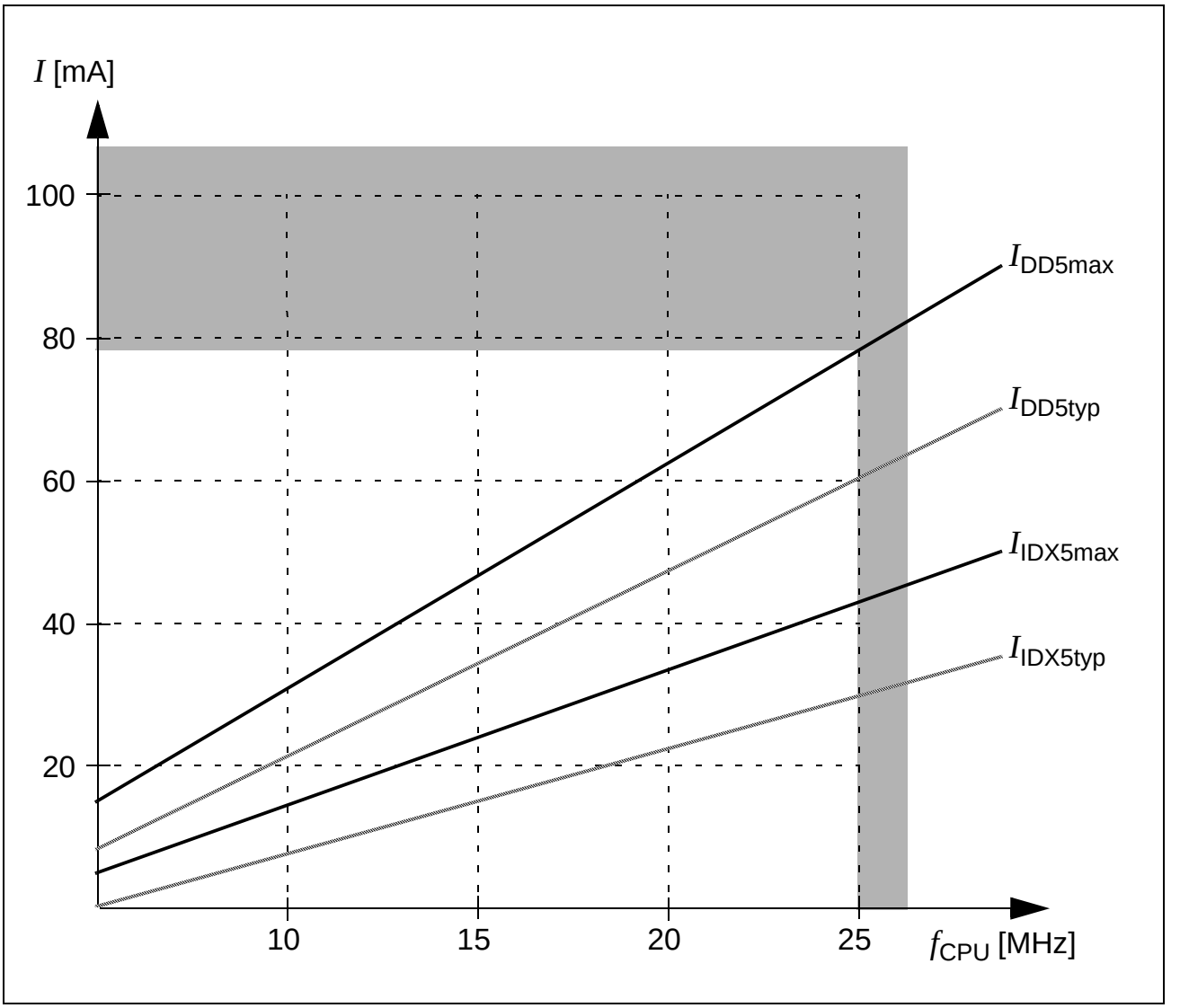

Figure 10 Supply/Idle Current as a Function of Operating Frequency

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:

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

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 $(3TCL)_{min} = 3TCL_{NOM} - 1.288$ 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**).*

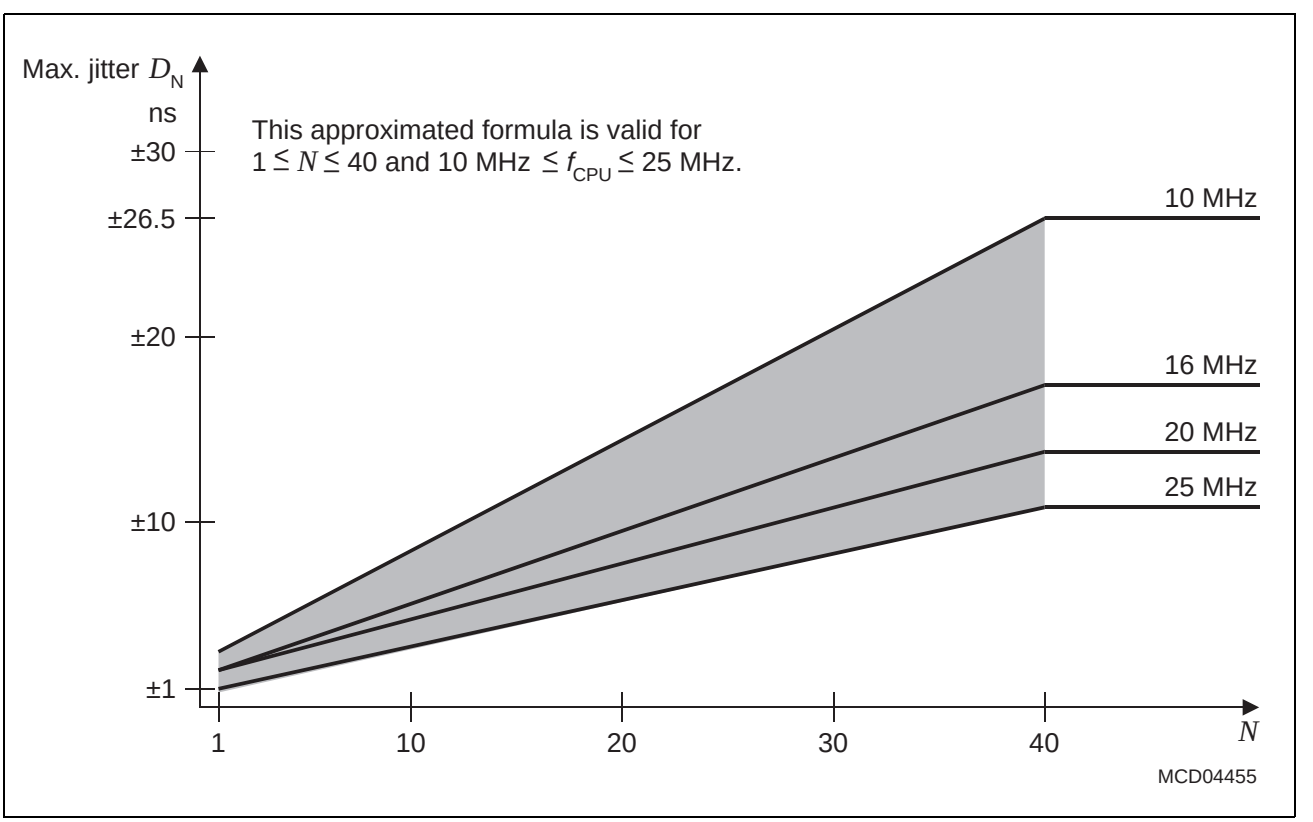

Figure 12 Approximated Maximum Accumulated PLL Jitter

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

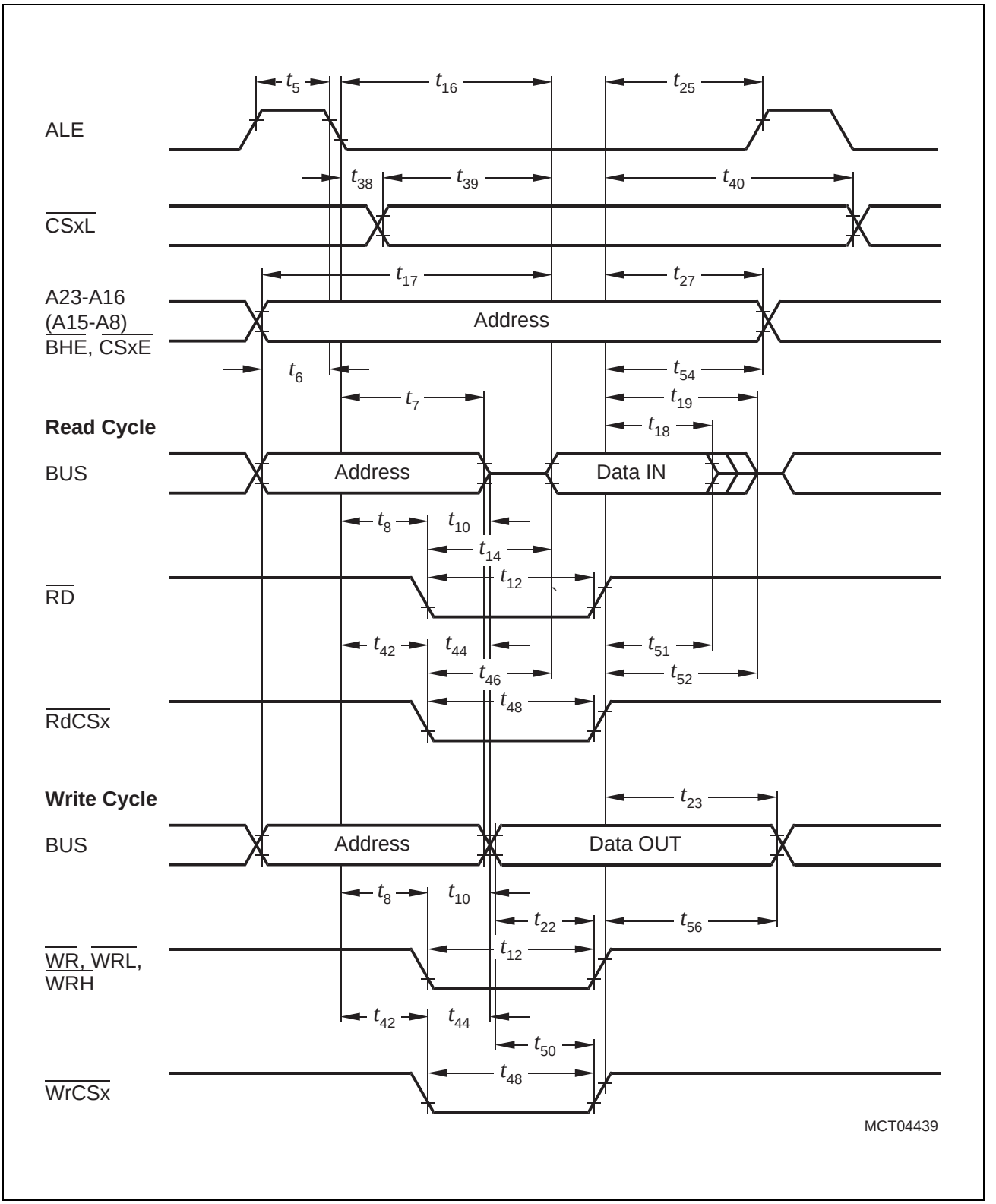

Figure 16 External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE

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

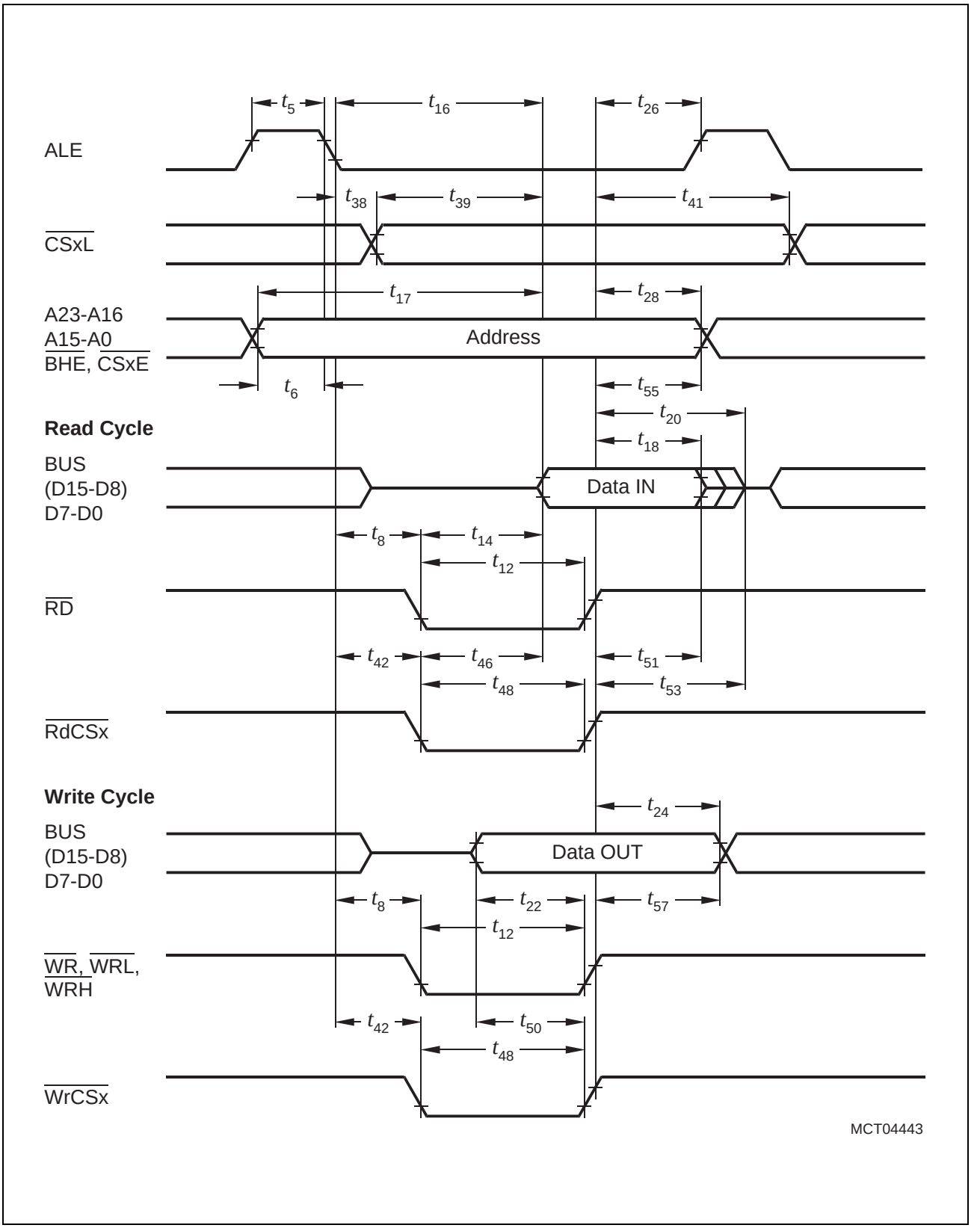

**Figure 20 External Memory Cycle:
Demultiplexed Bus, With Read/Write Delay, Normal ALE**

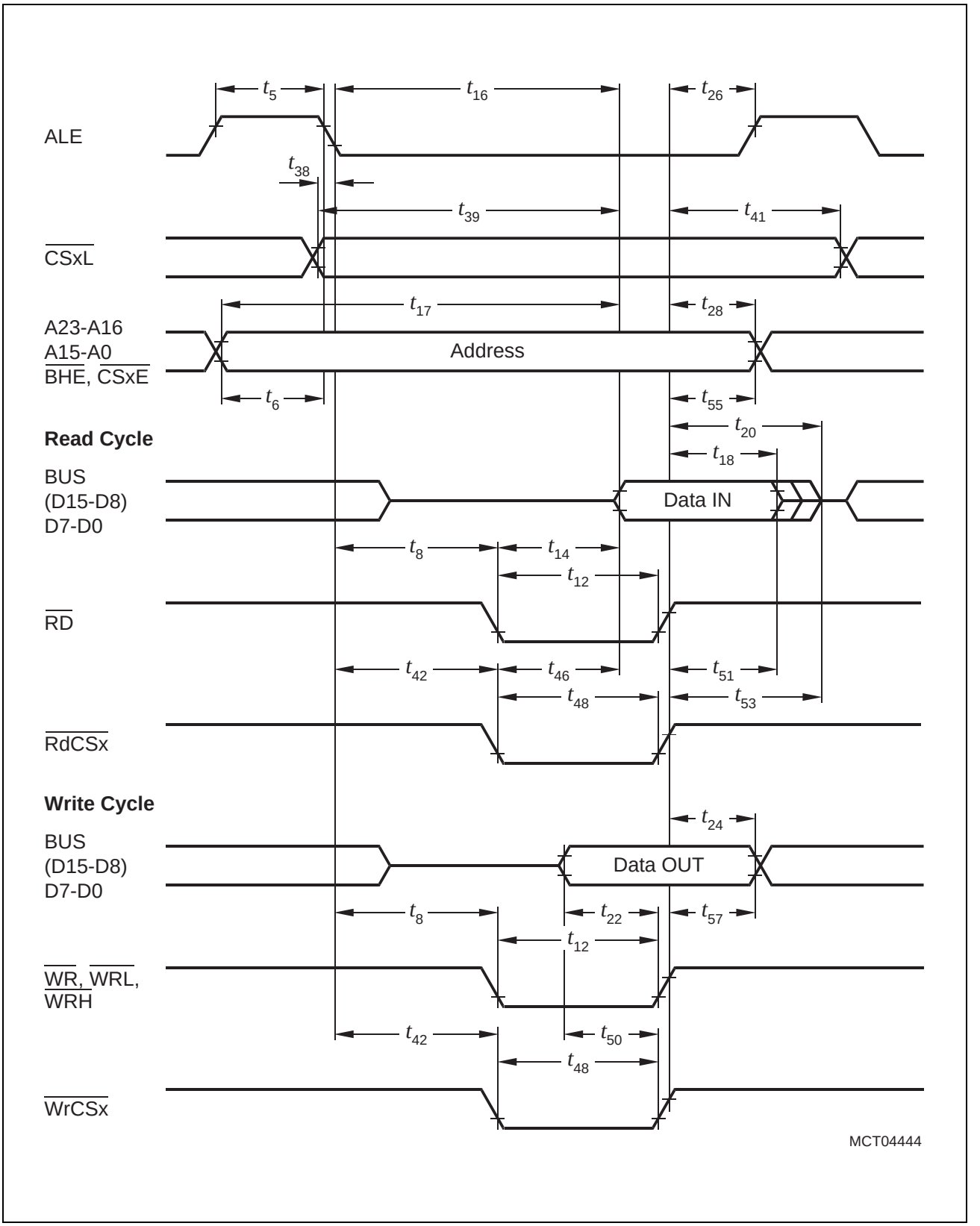

**Figure 21 External Memory Cycle:
Demultiplexed Bus, With Read/Write Delay, Extended ALE**

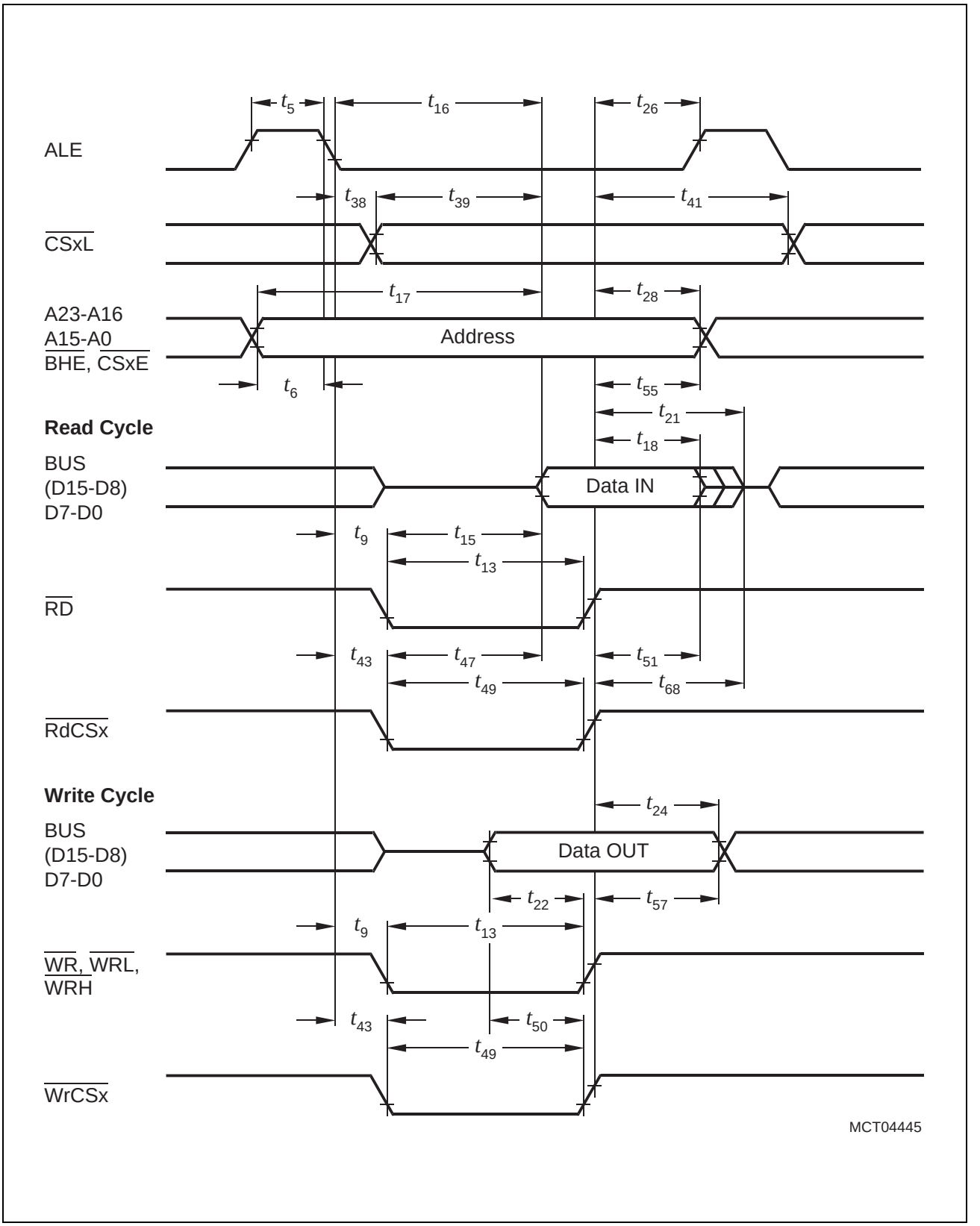

**Figure 22 External Memory Cycle:
Demultiplexed Bus, No Read/Write Delay, Normal ALE**

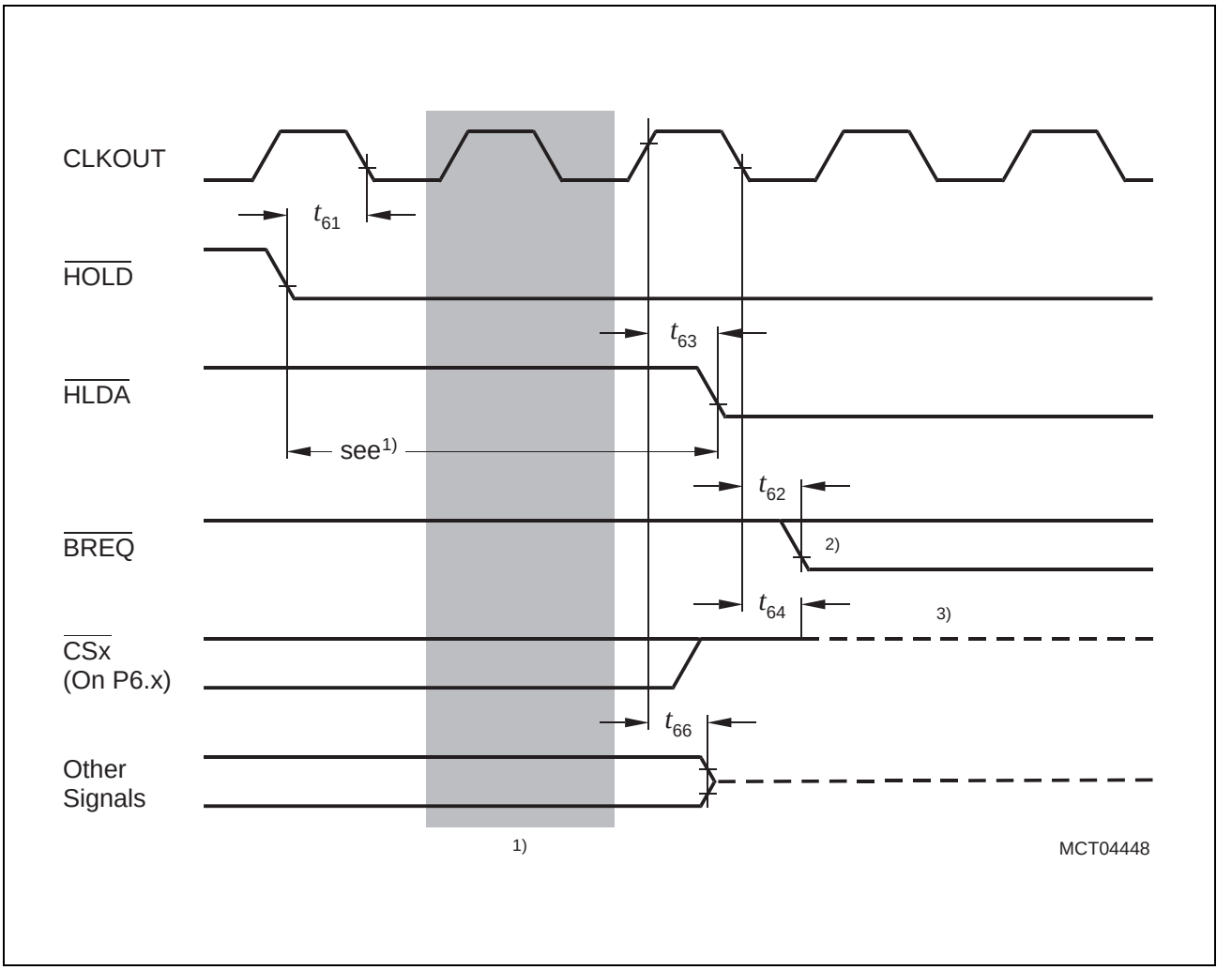

Figure 25 External Bus Arbitration, Releasing the Bus

Notes
1) The C161CS/JC/JI will complete the currently running bus cycle before granting bus access.
2) This is the first possibility for $\overline{\text{BREQ}}$ to get active.
3) The $\overline{\text{CS}}$ outputs will be resistive high (pullup) after t_{64}.