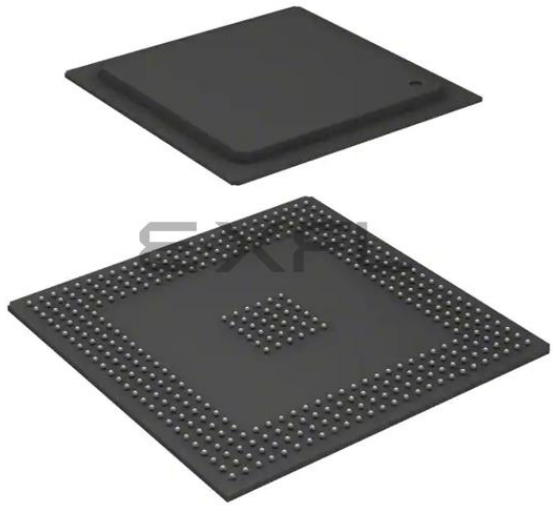


Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	PowerPC
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	56
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 2.7V
Data Converters	A/D 32x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	388-BBGA
Supplier Device Package	388-PBGA (27x27)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mpc563mvr66r

Contents

Chapter 21
CDR3 Flash (UC3F) EEPROM

Figures

provide maximum system safeguards against hardware and software faults. This chapter provides a detailed explanation of this functionality.

- Chapter 7, “Reset.” This section describes the MPC561/MPC563 reset sources, operation, control, and status.
- Chapter 8, “Clocks and Power Control,” describes the main timing and power control reference for the MPC561/MPC563.
- Chapter 9, “External Bus Interface,” describes the functionality of the MPC561/MPC563 external bus.
- Chapter 10, “Memory Controller,” generates interface signals to support a glueless interface to external memory and peripheral devices.
- Chapter 11, “L-Bus to U-Bus Interface (L2U),” describes the interface between the load/store bus (L-bus) and the unified bus (U-bus). The L2U module includes the Data Memory Protection Unit (DMPU), which provides protection for data memory accesses.
- Chapter 12, “U-Bus to IMB3 Bus Interface (UIMB).” The U-bus to IMB3 bus interface (UIMB) structure is used to connect the CPU internal unified bus (U-bus) to the intermodule bus 3 (IMB3). It controls bus communication between the U-bus and the IMB3.
- Chapter 13, “QADC64E Legacy Mode Operation.” The two queued analog-to-digital converter (QADC) modules on MPC561/MPC563 devices are 10-bit, unipolar, successive approximation converters. The modules can be configured to operate in one of two modes, legacy mode (MPC555 compatible) and enhanced mode. This chapter describes how the modules operate in legacy mode, which is the default mode of operation.
- Chapter 14, “QADC64E Enhanced Mode Operation.” The two queued analog-to-digital converter (QADC) modules on the MPC561/MPC563 devices are 10-bit, unipolar, successive approximation converters. The modules can be configured to operate in one of two modes, legacy mode (for MPC555 compatibility) and enhanced mode. This chapter describes how the module operates in enhanced mode.
- Chapter 15, “Queued Serial Multi-Channel Module.” The MPC561/MPC563 contains one queued serial multi-channel module (QSMCM) which provides three serial communication interfaces: the queued serial peripheral interface (QSPI) and two serial communications interfaces (SCI/UART). This chapter describes the functionality of each.
- Chapter 16, “CAN 2.0B Controller Module,” describes the three CAN 2.0B controller modules (TouCAN) implemented on the MPC561/MPC563. Each TouCAN is a communication controller that implements the Controller Area Network (CAN) protocol, an asynchronous communications protocol used in automotive and industrial control systems. It is a high speed (one Mbit/sec), short distance, priority based protocol that can run over a variety of mediums.
- Chapter 17, “Modular Input/Output Subsystem (MIOS14).” The modular I/O system (MIOS) consists of a library of flexible I/O and timer functions including I/O port, counters, input capture, output compare, pulse and period measurement, and PWM. Because the MIOS14 is composed of submodules, it is easily configurable for different kinds of applications.
- Chapter 18, “Peripheral Pin Multiplexing (PPM) Module.” The PPM functions as a parallel-to-serial communications module that reduces the number of signals required to connect

Table 2-14. MPC561/MPC563 Signal Reset State (continued)

Signal List[1]	Voltage	Slew Rate Controlled Option?	Drive Load (pF)[2]	Reset State	Hysteresis Enabled?	Function After HRESET, PORESET/TRST
B_AN3 / B_ANz / B_PQB3	5 V	Yes	NA	PU5 when driver not enabled or until PULL_DIS2 is set	No	B_AN3
	5 V	Yes	NA		No	
	5 V	Yes	50 ; 50 [5]		Yes	
B_AN[48:51]/ B_PQB[4:7]	5 V	Yes	NA	PU5 when driver not enabled or until PULL_DIS2 is set	No	B_AN[48:51]
	5 V	Yes	50 ; 50 [5]		Yes	
B_AN[52:54]/ B_MA[0:2]/ B_PQA[0:2]	5 V	Yes	NA	PU5 when driver not enabled or until PULL_DIS2 is set	No	B_AN[52:54]
	5 V	Yes	NA		Yes	
	5 V	Yes	50 ; 50 [5]		Yes	
B_AN[55:59]/ B_PQA[3:7]	5 V	Yes	NA	PU5 when driver not enabled or until PULL_DIS2 is set	No	B_AN[55:59]
	5 V	Yes	50 ; 50 [5]		Yes	
TouCAN_A / TouCAN_B						
A_CNTX0	5 V	Yes	50 ; 50 [17]	PU5 until PULL_DIS3 is set	No	A_CNTX0
B_CNTX0	5 V	Yes	50 ; 50 [17]	PU5 until PULL_DIS3 is set	No	B_CNTX0
A_CNRX0	5 V	No	NA	PU5 until PULL_DIS3 is set	Yes	A_CNRX0
B_CNRX0	5 V	No	NA	PU5 until PULL_DIS3 is set	Yes	B_CNRX0
UC3F Flash						
EPEE[18]	2.6 V	No	—	PU2.6	No	EPEE
B0EPEE[18]	2.6 V	No	—	PU2.6	No	B0EPEE
UC3F Power Supplies						
VFLASH [18]	5 V	—	—	—	—	VFLASH
VDDF [18]	2.6 V	—	—	—	—	VDDF
VSSF [18]	0 V	—	—	—	—	VSSF

Table 3-17. Instruction Set Summary (continued)

Mnemonic	Operand Syntax	Name
mullw (mullw. mullwo mullwo.)	rD,rA,rB	Multiply Low
nand (nand.)	rA,rS,rB	NAND
neg (neg. nego nego.)	rD,rA	Negate
nor (nor.)	rA,rS,rB	NOR
or (or.)	rA,rS,rB	OR
orc (orc.)	rA,rS,rB	OR with Complement
ori	rA,rS,UIMM	OR Immediate
oris	rA,rS,UIMM	OR Immediate Shifted
rfi	—	Return from Interrupt
rlwimi (rlwimi.)	rA,rS,SH,MB,ME	Rotate Left Word Immediate then Mask Insert
rlwinm (rlwinm.)	rA,rS,SH,MB,ME	Rotate Left Word Immediate then AND with Mask
rlwnm (rlwnm.)	rA,rS,rB,MB,ME	Rotate Left Word then AND with Mask
sc	—	System Call
slw (slw.)	rA,rS,rB	Shift Left Word
sraw (sraw.)	rA,rS,rB	Shift Right Algebraic Word
srawi (srawi.)	rA,rS,SH	Shift Right Algebraic Word Immediate
srw (srw.)	rA,rS,rB	Shift Right Word
stb	rS,d(rA)	Store Byte
stbu	rS,d(rA)	Store Byte with Update
stbux	rS,rA,rB	Store Byte with Update Indexed
stbx	rS,rA,rB	Store Byte Indexed
stfd	frS,d(rA)	Store Floating-Point Double
stfdu	frS,d(rA)	Store Floating-Point Double with Update
stfdux	frS,rB	Store Floating-Point Double with Update Indexed
stfdx	frS,rB	Store Floating-Point Double Indexed
stfiwx	frS,rB	Store Floating-Point as Integer Word Indexed
stfs	frS,d(rA)	Store Floating-Point Single
stfsu	frS,d(rA)	Store Floating-Point Single with Update
stfsux	frS,rB	Store Floating-Point Single with Update Indexed
stfsx	frS,r B	Store Floating-Point Single Indexed
sth	rS,d(rA)	Store Half-Word

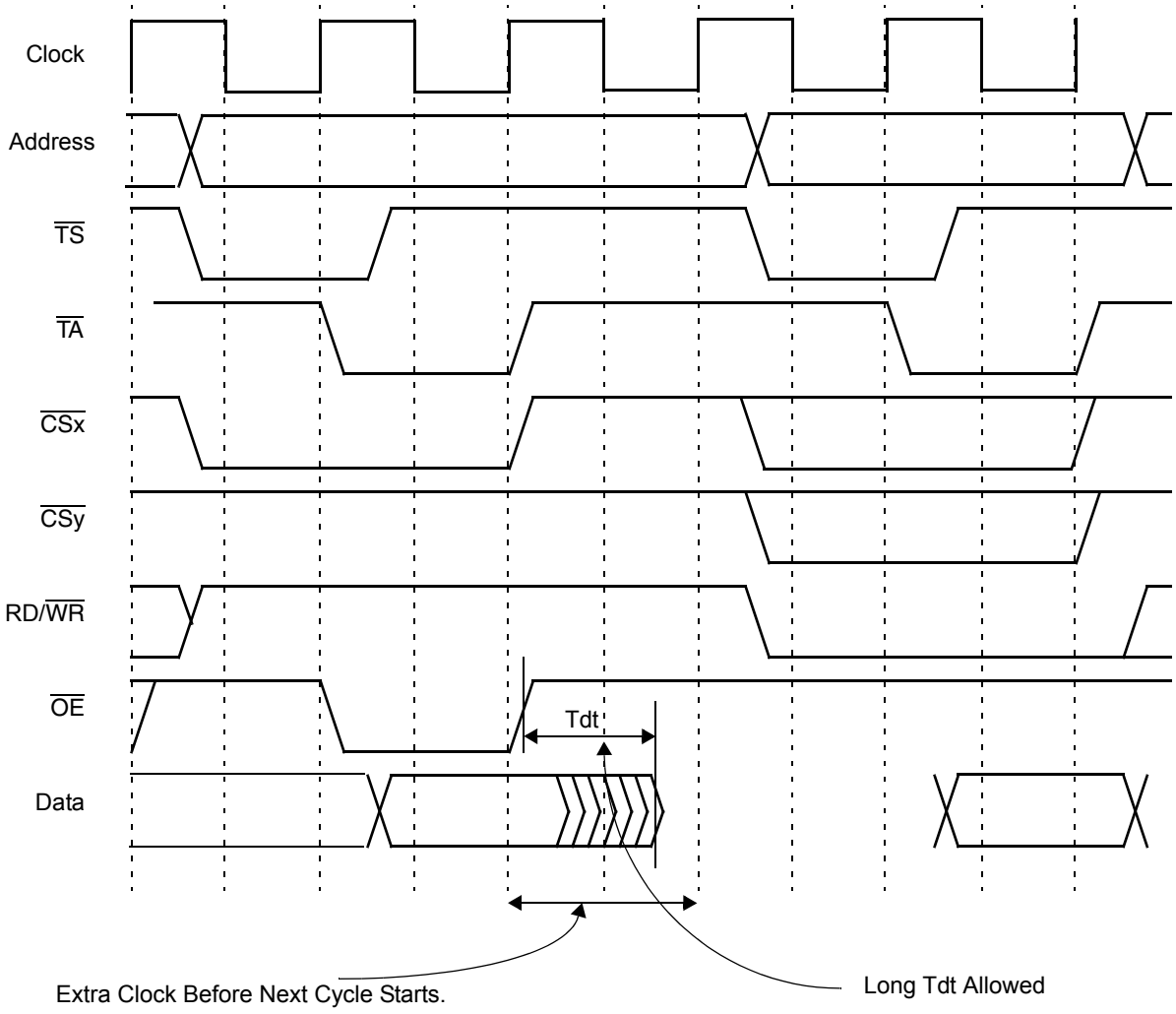


Figure 10-16. Consecutive Accesses (Write After Read, EHTR = 1)

Figure 10-17 shows consecutive accesses from different banks. Because EHTR = 1 (and the accesses are to different banks), an extra clock cycle is inserted.

Table 10-10. OR0–OR3 Bit Descriptions

Bits	Name	Description
0:16	AM	Address mask. This field allows masking of any corresponding bits in the associated base register. Masking the address bits independently allows external devices of different size address ranges to be used. Any clear bit masks the corresponding address bit. Any set bit causes the corresponding address bit to be used in comparison with the address signals. Address mask bits can be set or cleared in any order in the field, allowing a resource to reside in more than one area of the address map. This field can be read or written at anytime. Following a system reset, the AM bits are cleared in OR0.
17:19	ATM	Address type mask. This field masks selected address type bits, allowing more than one address space type to be assigned to a chip-select. Any set bit causes the corresponding address type code bits to be used as part of the address comparison. Any cleared bit masks the corresponding address type code bit. Clear the ATM bits to ignore address type codes as part of the address comparison. Note that the address type field uses only AT[0:2] and does not need AT3 to define the memory type space. Following a system reset, the ATM bits are cleared in OR0.
20	CSNT	Chip-select negation time. Following a system reset, the CSNT bit is reset in OR0. 0 $\overline{CS}$/$\overline{WE}$ are negated normally. 1 $\overline{CS}$/$\overline{WE}$ are negated a quarter of a clock earlier than normal Following a system reset, the CSNT bit is cleared in OR0.
21:22	ACS	Address to chip-select setup. Following a system reset, the ACS bits are reset in OR0. 00 $\overline{CS}$ is asserted at the same time that the address lines are valid. 01 Reserved 10 $\overline{CS}$ is asserted a quarter of a clock after the address lines are valid. 11 $\overline{CS}$ is asserted half a clock after the address lines are valid Following a system reset, the ACS bits are cleared in OR0.
23	EHTR	Extended hold time on read accesses. This bit, when asserted, inserts an idle clock cycle after a read access from the current bank and any MPC561/MPC563 write accesses or read accesses to a different bank. 0 Memory controller generates normal timing 1 Memory controller generates extended hold timing Following a system reset, the EHTR bits are cleared in OR0.
24:27	SCY	Cycle length in clocks. This four-bit value represents the number of wait states inserted in the single cycle, or in the first beat of a burst, when the GPCM handles the external memory access. Values range from 0 (0b0000) to 15 (0b1111). This is the main parameter for determining the length of the cycle. The total cycle length may vary depending on the settings of other timing attributes. The total memory access length is (2 + SCY) x Clocks. If an external $\overline{TA}$ response is selected for this memory bank (by setting the SETA bit), then the SCY field is not used. Following a system reset, the SCY bits are set to 0b1111 in OR0.

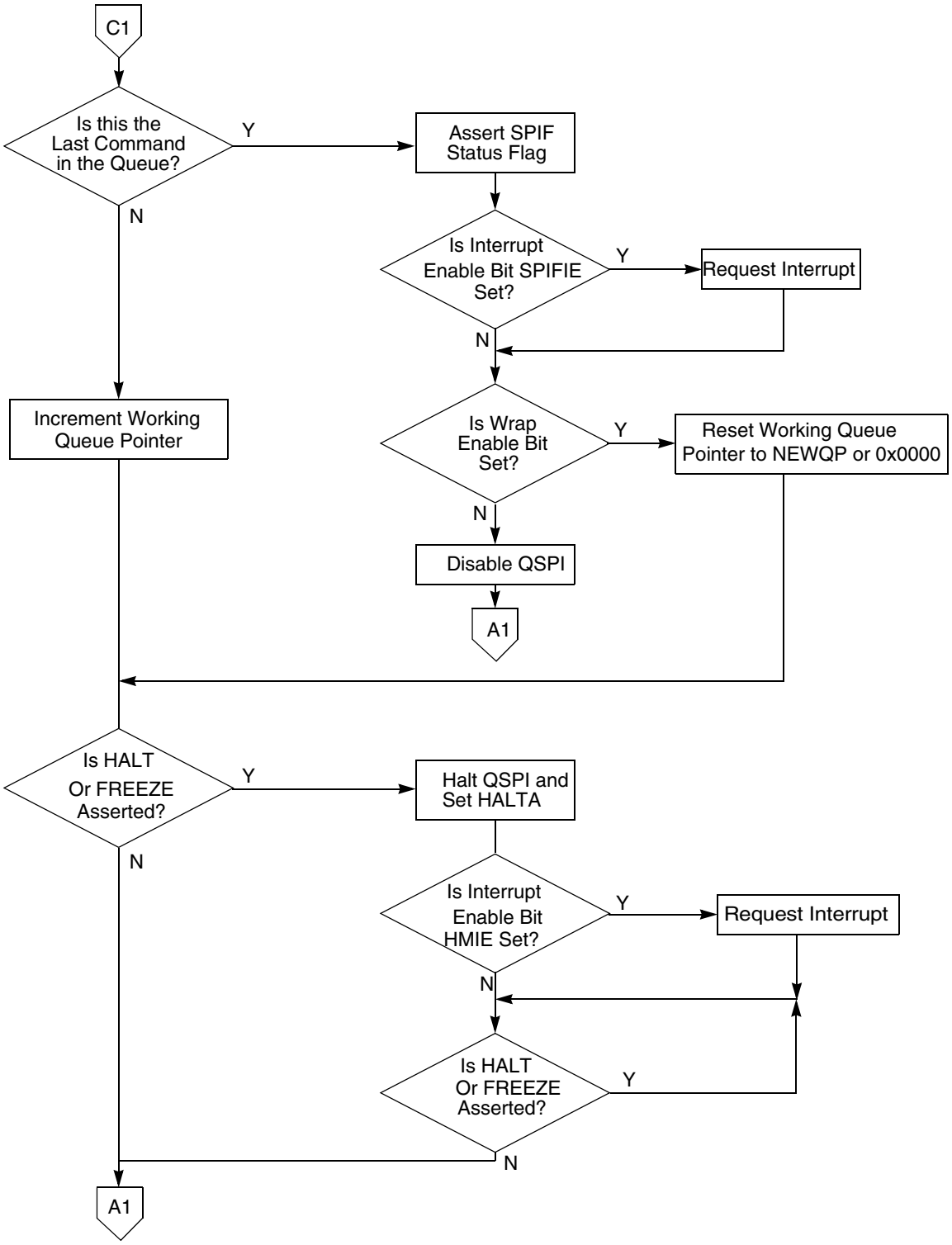


Figure 15-21. Flowchart of QSPI Master Operation (Part 3)

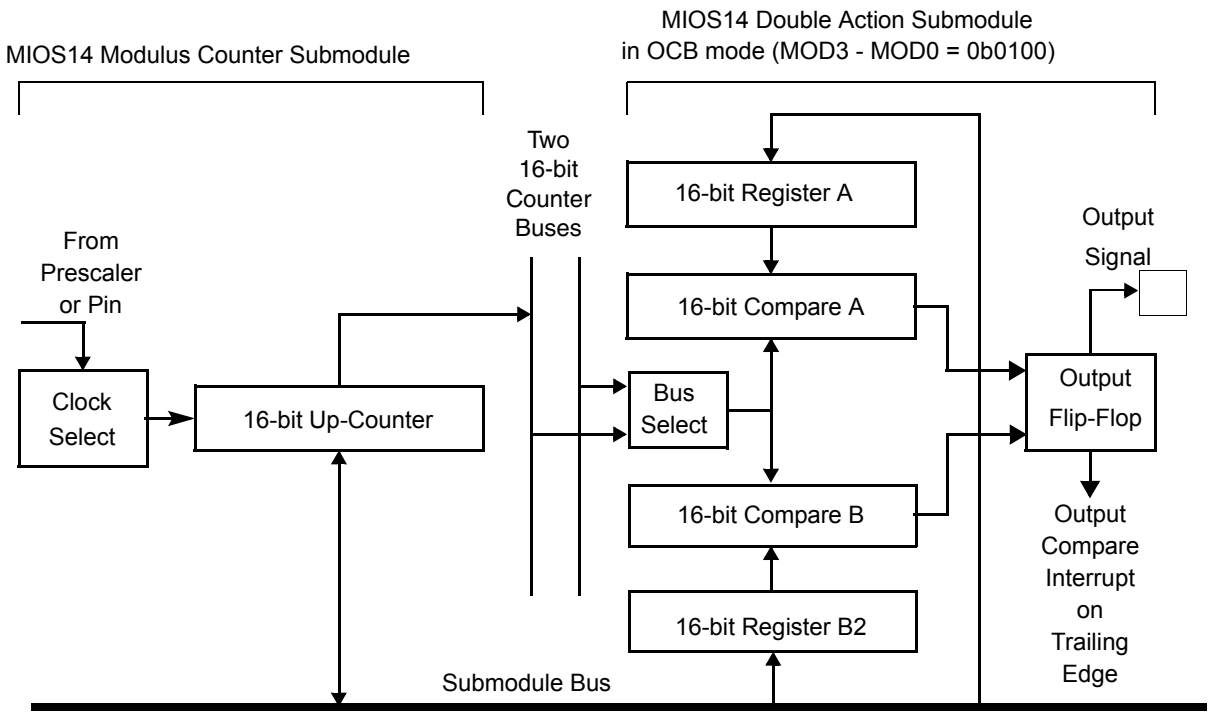
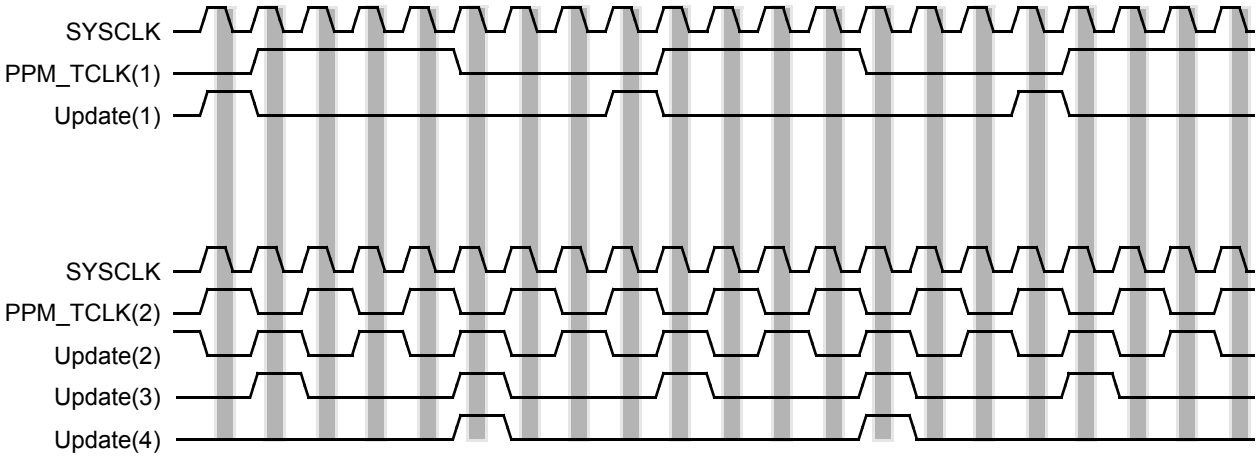


Figure 17-45. MIOS14 Example: Double Edge Output Compare

17.13.4 MIOS14 Output Pulse Width Modulation with MDASM

Output waveforms can be generated with any duty cycle without software involvement. The software sets up a MDASM with the compare times for the rising and falling edges and they are automatically repeated. The software does not need to respond to interrupts to generate continuous pulses. The frequency may be selected as the frequency of a free-running counter time-base, times a binary multiplier selected in the MDASM. Multiple PWM outputs can be created from multiple MDASMs and share one counter submodule, provided that the frequencies of all of the output signals are a binary multiple of the time-base and that the counter submodule is operating in a free-running mode. Each MDASM has a software selectable “don’t care” on high-order bits of the time-base comparison so that the frequency of one output can be a binary multiple of another signal. Masking the time-base serves to multiply the frequency of the time-base by a binary number to form the frequency of the output waveform. The duty cycle can vary from one cycle to 64-Kbyte cycles. The frequency can range from 0.48 Hz to 156 KHz, though the resolution decreases at the higher frequencies to as low as seven bits. The generation of output square wave signals is of course the special case where the high and low times are equal.

When an MMCSM is used to drive the time-base, the modulus value is the period of the output PWM signal. Figure 17-46 shows such an example. The polarity of the leading edge of an output waveform is programmable for a rising or a falling edge. The software selects the period of the output signal by programming the MMCSM with a modulus value. The leading edge compare value is written into register A by software and the trailing edge time is written into register B1. When the leading edge value is reached, the content of register B1 is transferred to register B2, to form the next trailing edge value. Subsequent changes to the output pulse width are made by writing a new time into register B1. Updates to the pulse width are always synchronized to the leading edge of the waveform.



PPM_TCLK(1) = SYSCLK/4

Update(1) every PPM_TCLK(1)clock

PPM_TCLK(2) = SYSCLK/2

Update(2) every PPM_TCLK(2) clock

Update(3) every second PPM_TCLK(2) clock

Update(4) every fourth PPM_TCLK(2) clock

Figure 18-7. Examples Of Several TCLK Frequencies and Sample Rates

The PPM Module has two data transmit signals, PPM_TX[0:1], and two data receive signals, PPM_RX[0:1]. The amount of data transferred on these signals depends on the setting in PPMPCR[OP_16_8]. If the PPM is configured to transfer data in 16 PPM_TCLK cycles per 16-bit word then all data in TX_DATA[0:15] is transmitted on the PPM_TX0 signal, and all data is received into RX_SHIFTER[0:15] from PPM_RX0. If the PPM is configured to transfer data in eight PPM_TCLK cycles per 16-bit word then the eight bits will transfer on each of the data transfer signals.

NOTE

Care must be taken when setting the sample rate with respect to the OP_16_8 bit setting. For example if the PPM is transferring data on an 8-clock cycle, then setting the sample rate to every 16 clocks will result in lost data.

In SPI mode the phase and polarity of PPM_TCLK is selectable by programming bits in the PPMPCR register. PPM_TCLK can have normal polarity (active high) or inverted polarity (active low). There are two clock phases available: valid data can be latched on the transition of PPM_TCLK from its active edge to inactive edge, or valid data can be latched on the transition of PPM_TCLK from its inactive edge to active edge.

See Section 18.4.2, “PPM Control Register (PPMPCR)” for more information on SPI mode PPM_TCLK settings.

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	LSB 15
Field	GPDO															
SRESET	0000_0000_0000_0000															
Addr	0x30 5C22															

Figure 18-21. General Purpose Data Out Register (GPDO)

18.4.9 General-Purpose Data In (GPDI)

GPDI is an internal register that receives data directly from the PPM input signals, PPM_TX[0:1]. By default, the receive configuration registers are set to direct received data from RX_DATA[0:15] to the GPDI[0:15] register.

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	LSB 15
Field	GPDI															
SRESET	0000_0000_0000_0000															
Addr	0x30 5C24															

Figure 18-22. General Purpose Data In Register (GPDI)

18.4.10 Short Register (SHORT_REG)

SHORT_REG allows the shorting of certain internal signals in the MPC561/MPC563 devices. This feature allows functions, whose internal signals are multiplexed on external signals, to be accessible simultaneously.

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	LSB 15
Field	SH_TCAN[2:0]			SH_TPU[1:0]		RESERVED		SH_ET1	SH_ET2	SH_T2CLK		RESERVED				
SRESET	0000_0000_0000_0000															
Addr	0x30 5C26															

Figure 18-23. Short Register (SHORT_REG)

instruction. Therefore, a valid data status will be output and the interrupt status will be saved for the next transmission.

The sequencing error encoding indicates that the inputs from the external development tool are not what the development port and/or the CPU was expecting. Two cases could cause this error:

1. The processor was trying to read instructions and there was data shifted into the development port, or
2. The processor was trying to read data and there was instruction shifted into the development port. The port will terminate the read cycle with a bus error.

This bus error will cause the CPU to signal that an interrupt (exception) occurred. Since a status of sequencing error has a higher priority than exception, the port will report the sequencing error first, and the CPU interrupt on the next transmission. The development port will ignore the command, instruction, or data shifted in while the sequencing error or CPU interrupt is shifted out. The next transmission after all error status is reported to the port should be a new instruction, trap enable or command (possibly the one that was in progress when the sequencing error occurred).

The interrupt-occurred encoding is used to indicate that the CPU encountered an interrupt during the execution of the previous instruction in debug mode. Interrupts may occur as the result of instruction execution (such as unimplemented opcode or arithmetic error), because of a memory access fault, or from an unmasked external interrupt. When an interrupt occurs the development port will ignore the command, instruction, or data shifted in while the interrupt encoding was shifting out. The next transmission to the port should be a new instruction, trap enable or debug port command.

Finally, the null encoding is used to indicate that no data has been transferred from the CPU to the development port shift register.

23.4.6.11 Fast Download Procedure

The download procedure is used to download a block of data from the debug tool into system memory. This procedure can be accomplished by repeating the following sequence of transactions from the development tool to the debug port for the number of data words to be down loaded:

```
INIT:    Save RX, RY
         RY <- Memory Block address- 4

         …

repeat:  mfspr    RX, DPDR
         DATA word to be moved to memory
         stwu     RX, 0x4(RY)
until here

         …

         Restore RX,RY
```

Figure 23-12. Download Procedure Code Example

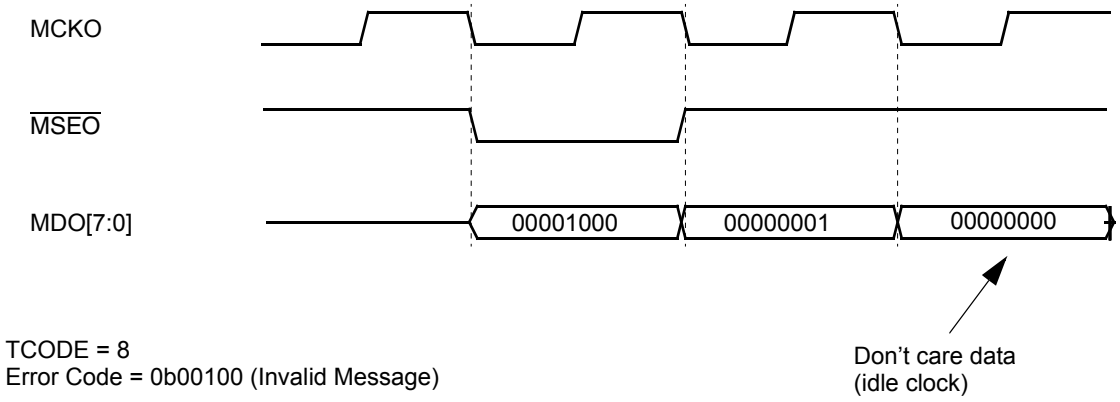


Figure 24-86. Error Message (Invalid Message)

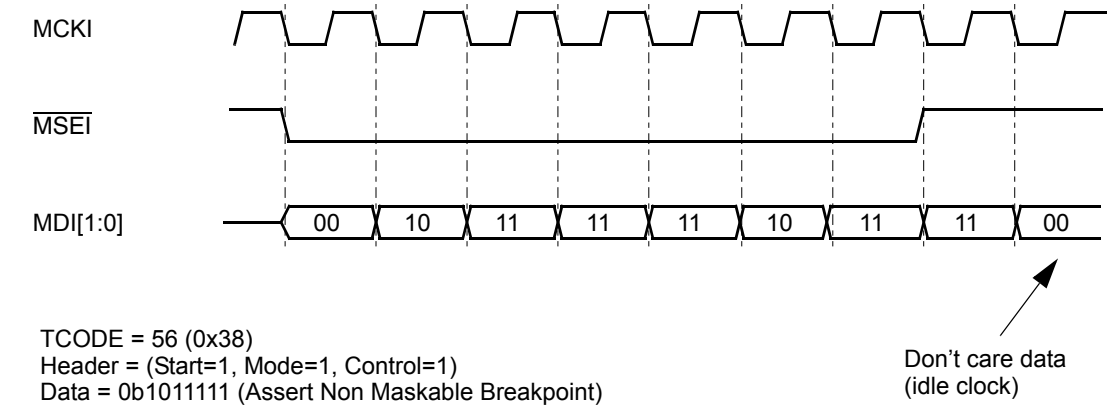


Figure 24-87. DSDI Data Message (Assert Non-Maskable Breakpoint)

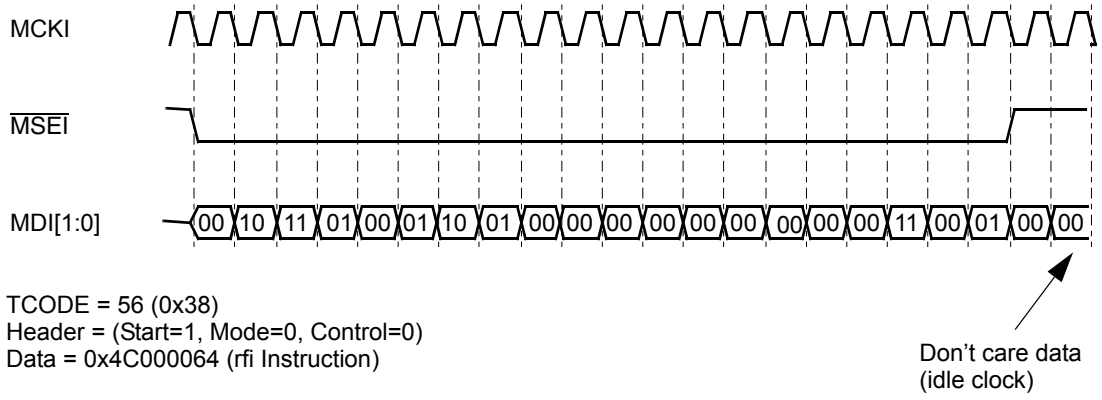


Figure 24-88. DSDI Data Message (CPU Instruction — rfi)

Table 25-1. MPC561 Boundary Scan Bit Definition (continued)

BSDL Bit	Cell Type	Pin/Port Name	BSDL Function	Safe Value	Control Cell	Disable Value	Disable Result	Pin Function	Pad Type
26	BC_2	*	controlr	0					
27	BC_7	B_TPUCH11	bidir	0	26	0	Z	IO	5vsa
28	BC_2	*	controlr	0					
29	BC_7	B_TPUCH12	bidir	0	28	0	Z	IO	5vsa
30	BC_2	*	controlr	0					
31	BC_7	B_TPUCH13	bidir	0	30	0	Z	IO	5vsa
32	BC_2	*	controlr	0					
33	BC_7	B_TPUCH14	bidir	0	32	0	Z	IO	5vsa
34	BC_2	*	controlr	0					
35	BC_7	B_TPUCH15	bidir	0	34	0	Z	IO	5vsa
36	BC_2	*	controlr	0					
37	BC_7	B_T2CLK_PCS4	bidir	0	36	0	Z	IO	5vfa
38	BC_2	*	controlr	0					
39	BC_7	A_T2CLK_PCS5	bidir	0	38	0	Z	IO	5vfa
40	BC_2	*	controlr	0					
41	BC_7	A_TPUCH0	bidir	0	40	0	Z	IO	5vsa
42	BC_2	*	controlr	0					
43	BC_7	A_TPUCH1	bidir	0	42	0	Z	IO	5vsa
44	BC_2	*	controlr	0					
45	BC_7	A_TPUCH2	bidir	0	44	0	Z	IO	5vsa
46	BC_2	*	controlr	0					
47	BC_7	A_TPUCH3	bidir	0	46	0	Z	IO	5vsa
48	BC_2	*	controlr	0					
49	BC_7	A_TPUCH4	bidir	0	48	0	Z	IO	5vsa
50	BC_2	*	controlr	0					
51	BC_7	A_TPUCH5	bidir	0	50	0	Z	IO	5vsa
52	BC_2	*	controlr	0					
53	BC_7	A_TPUCH6	bidir	0	52	0	Z	IO	5vsa
54	BC_2	*	controlr	0					
55	BC_7	A_TPUCH7	bidir	0	54	0	Z	IO	5vsa
56	BC_2	*	controlr	0					
57	BC_7	A_TPUCH8	bidir	0	56	0	Z	IO	5vsa
58	BC_2	*	controlr	0					
59	BC_7	A_TPUCH9	bidir	0	58	0	Z	IO	5vsa
60	BC_2	*	controlr	0					
61	BC_7	A_TPUCH10	bidir	0	60	0	Z	IO	5vsa

3. Compression of one of the instruction's halves into a vocabulary pointer and bypass of the other half. A bypassed field is one for which non-compressed data (16-bit halfword or 32-bit word) is placed in the compressed code. After compression is defined, the non-compressed data field is defined in the class.
4. Bypass of the whole instruction. No compression is permitted.

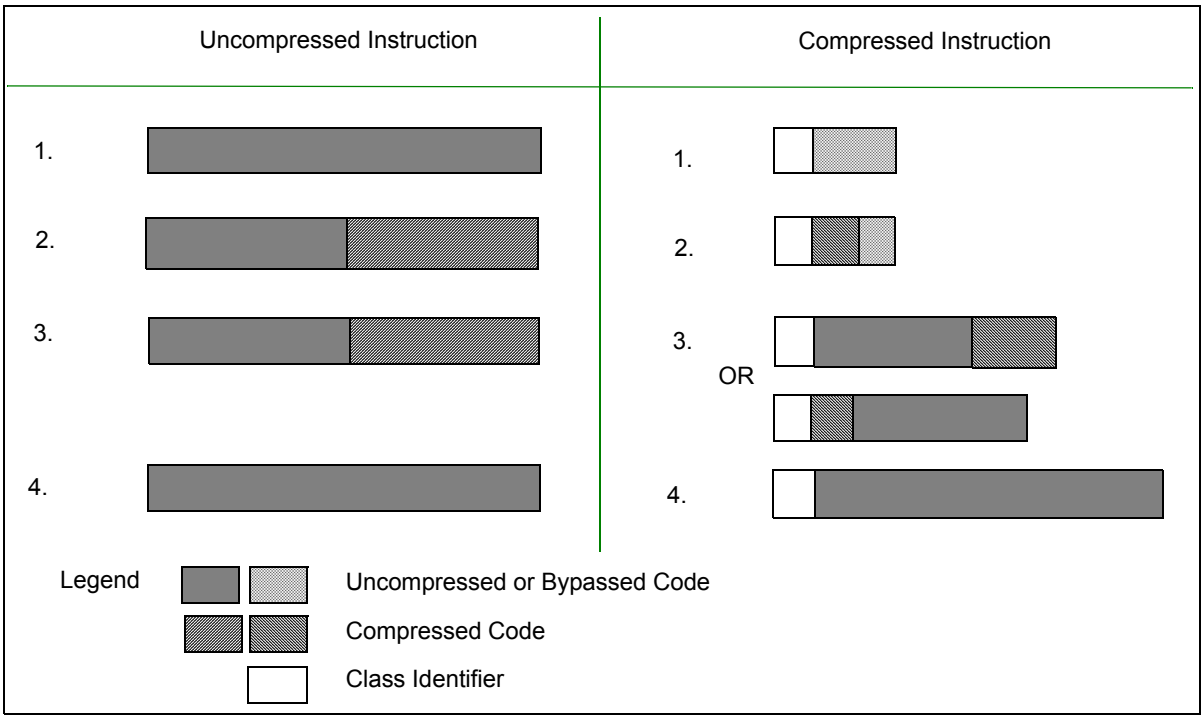


Figure A-1. Instruction Compression Alternatives

A 4-bit class identifier is added to the beginning of each compressed instruction to supply class identification during decompression. Compressed and bypass field lengths may vary. (A fully bypassed instruction, including its 4-bit class identifier, is 36 bits.)

The compressed instruction is guaranteed to start on an even bit. Thus, four bits are needed to find the starting location of the instruction inside a memory word. The instruction address in decompression on mode consists of a 28-bit word address (1 Gbyte of address space) and a 4-bit instruction pointer (IP). See Figure A-2.

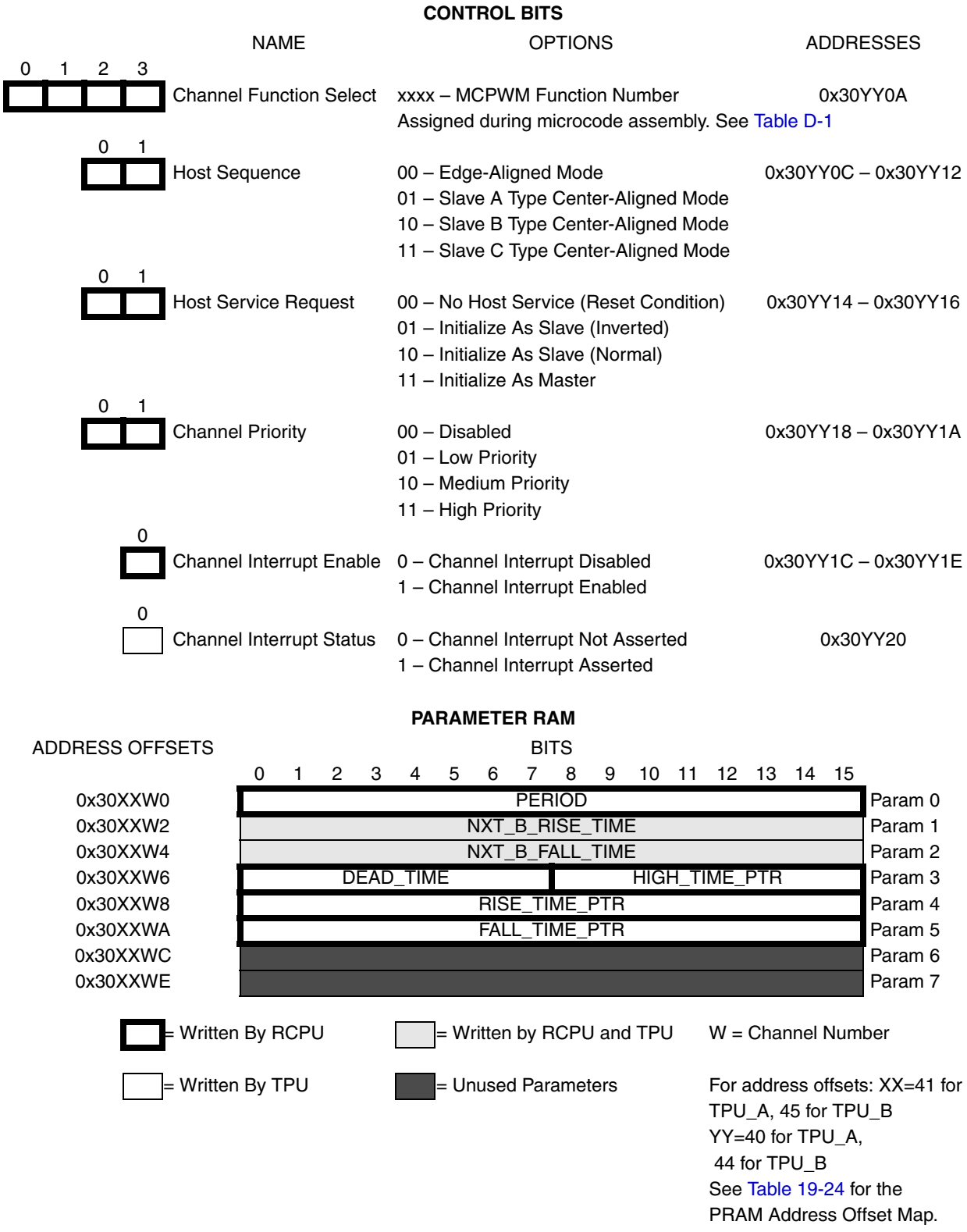


Figure D-16. MCPWM Parameters — Slave Ch A Inverted Center-Aligned Mode

Table F-10. Bus Operation Timing (continued)

Note: (V_{DD} = 2.6 V ± 0.1 V, V_{DDH} = 5.0 V ± 0.25 V, T_A = T_L to T_H, 50 pF load unless noted otherwise)

	Characteristic	40 MHz		56 MHz[1]		Unit
		Min	Max	Min	Max	
29	$\overline{TS}$ valid to CLKOUT Rising Edge (Setup Time)	7		5		ns
30	CLKOUT Rising Edge to $\overline{TS}$ Valid (Hold Time).	5		5		ns

1 56-MHz operation is available as an option. Some parts (without the 56-MHz option) will operate at a maximum frequency of 40 MHz.

2 The timing for $\overline{BR}$ output is relevant when the MPC561/MPC563 is selected to work with external bus arbiter. The timing for $\overline{BG}$ output is relevant when the MPC561/MPC563 is selected to work with internal bus arbiter.

3 The setup times required for $\overline{TA}$, $\overline{TEA}$, and $\overline{BI}$ are relevant only when they are supplied by the external device (and not the memory controller).

4 The maximum value of spec 8 for DATA[0:31] pins must be extended by 1.1 ns if the pins have been precharged to greater than V_{DDL}. This is the case if an external slave device on the bus is running at the max. value of VDATAPC. This is currently specified at 3.1 V. The 1.1 ns addition to spec 8 reflects the expected timing degradation for 3.1 V.

5 The timing 27 refers to $\overline{CS}$ when ACS = '00' and to $\overline{WE}$[0:3]/$\overline{BE}$[0:3] when CSNT = '0'.

NOTE

The D[0:31] input timings 17 and 18 refer to the rising edge of the CLKOUT in which the $\overline{TA}$ input signal is asserted.

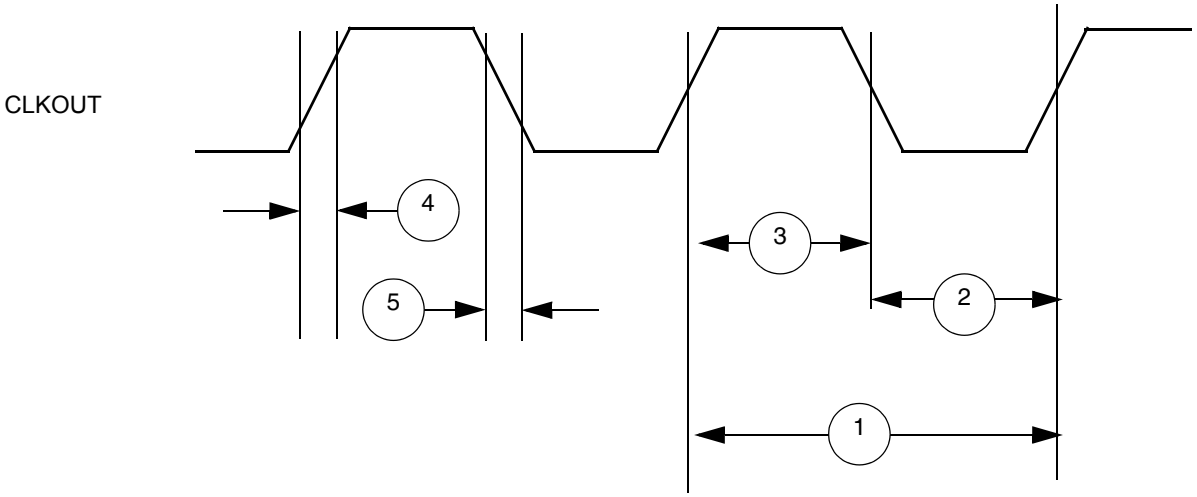


Figure F-10. CLKOUT Pin Timing

Table G-10. Bus Operation Timing (continued)

Note: (V_{DD} = 2.6 V ± 0.1 V, V_{DDH} = 5.0 V ± 0.25 V, $T_A = T_L$ to T_H, 50 pF load unless noted otherwise)

	Characteristic	66 MHz		Unit
		Min	Max	
19b	CLKOUT Falling Edge to $\overline{CS}$ asserted -GPCM- ACS = 11, TRLX = 0 or 1	4	9	ns
19c	CLKOUT Falling Edge to $\overline{CS}$ asserted -GPCM- ACS = 11, TRLX = 0, EBDF = 1	6.69	12.69	ns
20	CLKOUT Rising Edge to $\overline{CS}$ negated -GPCM- Read Access or Write access when CSNT = 0 or write access when CSNT = 1 and ACS = 00	1.55	4.85	ns
21	ADDR[8:31] to $\overline{CS}$ asserted -GPCM- ACS = 10, TRLX = 0	1.2	—	ns
21a	ADDR[8:31] to $\overline{CS}$ asserted -GPCM- ACS = 11, TRLX = 0	5.1	—	ns
22	CLKOUT Rising Edge to $\overline{OE}$, $\overline{WE}$[0:3]/$\overline{BE}$[0:3] asserted	1	5.45	ns
23	CLKOUT Rising Edge to $\overline{OE}$ negated	1.45	5.06	ns
24	ADDR[8:31] to $\overline{CS}$ asserted -GPCM- ACS = 10, TRLX = 1	13.95	—	ns
24a	ADDR[8:31] to $\overline{CS}$ asserted -GPCM- ACS = 11, TRLX = 1	17	—	ns
25	CLKOUT Rising Edge to $\overline{WE}$[0:3]/$\overline{BE}$[0:3] negated -GPCM-write access CSNT = ‘0‘	—	4.75	ns
25a	CLKOUT Falling Edge to $\overline{WE}$[0:3]/$\overline{BE}$[0:3] negated -GPCM-write access TRLX = ‘0’ or ‘1’, CSNT = ‘1, EBDF = 0’.	4.5	9.5	ns
25b	CLKOUT Falling Edge to $\overline{CS}$ negated -GPCM-write access TRLX = ‘0’ or ‘1’, CSNT = ‘1’, ACS = ‘10’ or ACS=’11’, EBDF = 0	4.5	9.5	ns
25c	CLKOUT Falling Edge to $\overline{WE}$[0:3]/$\overline{BE}$[0:3] negated -GPCM-write access TRLX = ‘0’, CSNT = ‘1, EBDF = 1’.	5.5	12.69	ns
25d	CLKOUT Falling Edge to $\overline{CS}$ negated -GPCM-write access TRLX = ‘0’, CSNT = ‘1’, ACS = ‘10’ or ACS=’11’, EBDF = 1	6.25	17	ns
26	$\overline{WE}$[0:3]/$\overline{BE}$[0:3] negated to D[0:31] High Z -GPCM- write access, CSNT = ‘0’	1.95	—	ns
26a	$\overline{WE}$[0:3]/$\overline{BE}$[0:3] negated to D[0:31] High Z -GPCM- write access, TRLX = ‘0’, CSNT = ‘1’, EBDF = 0	4.85	—	ns
26b	$\overline{CS}$ negated to D[0:31], High Z -GPCM- write access, ACS = ‘00’, TRLX = ‘0’ & CSNT = ‘0’	1.95	—	ns