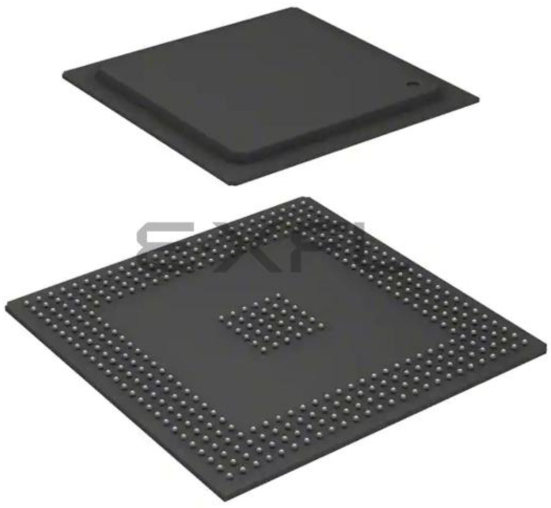


What is "Embedded - Microcontrollers"?

"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	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 ~ 85°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/mpc564cvr66

Contents

Table 3-5. FPSCR Bit Descriptions (continued)

Bits	Name	Description	
12	VXVC	Floating-point invalid operation exception for invalid compare.	Sticky bit
13	FR	Floating-point fraction rounded. The last floating-point instruction that potentially rounded the intermediate result incremented the fraction.	Not sticky
14	FI	Floating-point fraction inexact. The last floating-point instruction that potentially rounded the intermediate result produced an inexact fraction or a disabled exponent overflow.	Not sticky
[15:19]	FPRF	Floating-point result flags. This field is based on the value placed into the target register even if that value is undefined. Refer to Table 3-6 for specific bit settings. 15 Floating-point result class descriptor (C). Floating-point instructions other than the compare instructions may set this bit with the FPCC bits, to indicate the class of the result. 16-19 Floating-point condition code (FPCC). Floating-point compare instructions always set one of the FPCC bits to one and the other three FPCC bits to zero. Other floating-point instructions may set the FPCC bits with the C bit, to indicate the class of the result. Note that in this case the high-order three bits of the FPCC retain their relational significance indicating that the value is less than, greater than, or equal to zero. 16 Floating-point less than or negative (FL or <) 17 Floating-point greater than or positive (FG or >) 18 Floating-point equal or zero (FE or =) 19 Floating-point unordered or NaN (FU or ?)	Not sticky
20	—	Reserved	—
21	VXSOFT	Floating-point invalid operation exception for software request. This bit can be altered only by the mcrfs, mtfsfi, mtfsf, mtfsb0, or mtfsb1 instructions. The purpose of VXSOFT is to allow software to cause an invalid operation condition for a condition that is not necessarily associated with the execution of a floating-point instruction. For example, it might be set by a program that computes a square root if the source operand is negative.	Sticky bit
22	VXSQRT	Floating-point invalid operation exception for invalid square root. This guarantees that software can simulate fsqrt and frsqrte, and can provide a consistent interface to handle exceptions caused by square root operations.	Sticky bit
23	VXCVI	Floating-point invalid operation exception for invalid integer convert.	Sticky bit
24	VE	Floating-point invalid operation exception enable.	—
25	OE	Floating-point overflow exception enable.	—
26	UE	Floating-point underflow exception enable. This bit should not be used to determine whether denormalization should be performed on floating-point stores.	—
27	ZE	Floating-point zero divide exception enable.	—
28	XE	Floating-point inexact exception enable.	—

Section 4.6.2.3, "Region Attribute Registers (MI_RA[0:3])," and Section 4.6.2.4, "Global Region Attribute Register (MI_GRA)" for details.

4.6 BBC Programming Model

4.6.1 Address Map

The BBC consists of three separately addressable sections within the internal chip address space:

1. BBC and IMPU control registers. These are mapped in the SPR registers area and may be programmed by using the RCPU mtspr/mfspr instructions.
2. Decompressor vocabulary RAM (DECRAM). The DECRAM array occupies the 2-Kbyte physical memory (8 Kbytes of the MPC561/MPC563 address space is allocated for DECRAM).
3. Decompressor class configuration registers (DCCR) block. It consists of 15 decompression class configuration registers. These registers are available for word wide read/write accesses through U-bus. The registers occupy a 64-byte physical block (8-Kbyte chip address space is allocated for the register block).

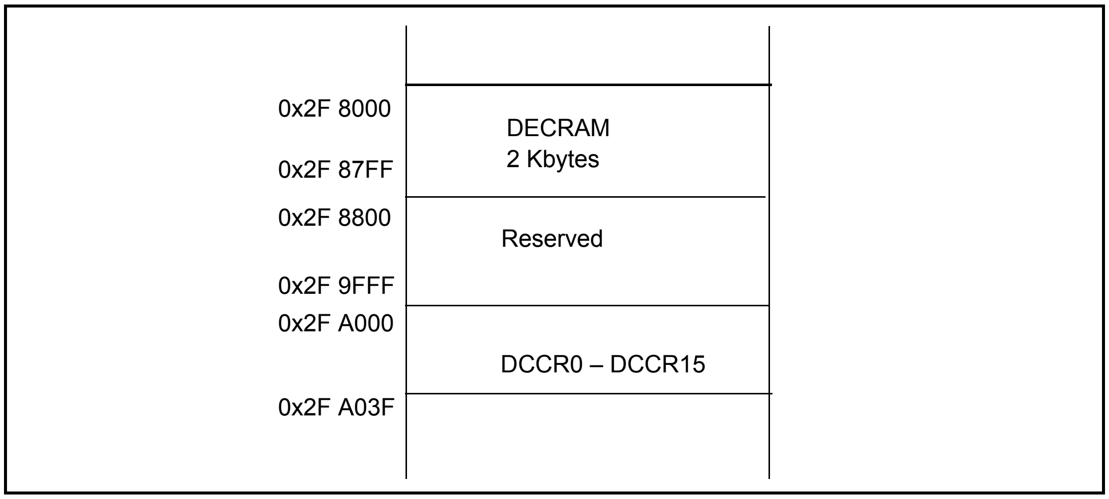


Figure 4-6. MPC561/MPC563 Memory Map

4.6.1.1 BBC Special Purpose Registers (SPRs)

Table 4-3. BBC SPRs

SPR Number (Decimal)	Address for External Master Access (Hex)	Register Name
528	0x2100	IMPU Global Region Attribute Register (MI_GRA). See Table 4-8 for bits descriptions.
529	0x2300	External Interrupt Relocation Table Base Address Register (EIBADR). See Table 4-9 for bits descriptions.
560	0x2110	BBC Module Configuration Register (BBCMCR). See Table 4-4 for bits descriptions

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	LSB 15
Field	SWSR															
Reset	0000_0000_0000_0000															
Addr	0x2F C00E															

Figure 6-27. Software Service Register (SWSR)

Table 6-16. SWSR Bit Descriptions

Bits	Name	Description
0:15	SWSR	SWT servicing sequence is written to this register. To prevent SWT time-out, a 0x556C followed by 0xAA39 should be written to this register. The SWSR can be written at any time but returns all zeros when read.

6.2.2.3.3 Transfer Error Status Register (TESR)

The transfer error status register contains a bit for each exception source generated by a transfer error. A bit set to logic 1 indicates what type of transfer error exception occurred since the last time the bits were cleared by reset or by the normal software status bit-clearing mechanism.

NOTE

These bits may be set due to canceled speculative accesses which do not cause an interrupt. The register has two identical sets of bit fields; one is associated with instruction transfers and the other with data transfers.

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	—															
Reset	0000_0000_0000_0000															
Addr	0x2F C020															

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	LSB 31
Field	—		IEXT	IBMT	—						DEXT	DBM	—			
Reset	0000_0000_0000_0000															

Figure 6-28. Transfer Error Status Register (TESR)

Table 6-17. TESR Bit Descriptions

Bits	Name	Description
0:17	—	Reserved
18	IEXT	Instruction external transfer error acknowledge. This bit is set if the cycle was terminated by an externally generated $\overline{\text{TEA}}$ signal when an instruction fetch was initiated.
19	IBMT	Instruction transfer monitor time out. This bit is set if the cycle was terminated by a bus monitor time-out when an instruction fetch was initiated.

For addition details, refer to Section 9.5.4, “Burst Transfer."

10.2.6 Reduced Data Setup Time

In order to meet timing requirements when interfacing to external memories, the data setup time can be reduced. This mode can be selected by programming the BR*x* registers. Thus there is flexibility in how each region can be configured to operate. The operation mode will be determined dynamically according to a particular access type. This means that for a memory region with the reduced setup time mode enabled, the mode will automatically switch to disabled when there is no requirement for the reduced setup time, (e.g., a back-to-back load/store access). For a new access with burst length more than 1, the operation mode will be automatically switched back to the reduced setup time mode.

Reduced setup time can be selected via the SST bit in BR[0:3]. See Section 10.9.3, “Memory Controller Base Registers (BR0–BR3)” for more details. If SCCR[EBDF] is greater than 0, however, an external burst access with reduced data setup time will corrupt a load/store to any USIU register.

The reduced setup time mode may or may not have a performance impact, depending on the properties of the memory. Namely, there is always an additional empty cycle between two burst sequences. On the other hand, this additional cycle, under certain conditions, may be compensated for by reducing the number of cycles in initial data access and sequential burst beats.

Table 10-1. Timing Requirements for Reduced Setup Time

CPU Specification	Memory Device Requirements
Cycle time at 56 MHz = 17.9 ns	Initial access time = 49 ns
Short setup time = 3 ns	Burst access time = 13 ns
Normal setup time = 6 ns	
Additional delay arising from on-board wires and clock skew between internal clock and CLKOUT	

10.2.6.1 Case 1: Normal Setup Time

Initial access:

$$\text{Initial access time of memory} + \text{Data setup time of CPU} + \text{Delays} = 49 + 6 + 1 = 56\text{ns}$$

To derive the number of clocks required, divide by the system clock cycle time:

$$\frac{56}{17.9} = 3.13 \qquad \text{therefore 4 cycles are required}$$

Burst access:

$$\text{Burst access time of memory} + \text{Data setup time of CPU} + \text{Delays} = 13 + 6 + 1 = 20\text{ns}$$

The number of clocks required $= \frac{20}{17.9} = 1.11$ therefore 2 clocks are required.

This case is illustrated in Figure 10-5.

generated internally and the QADC64E immediately begins execution of the first CCW in the queue. If a pause occurs, another trigger event is generated internally, and then execution continues without pausing.

The QADC64E automatically performs the conversions in the queue until an end-of-queue condition is encountered. The queue remains idle until the software again sets the single-scan enable bit. While the time to internally generate and act on a trigger event is very short, software can momentarily read the status conditions, indicating that the queue is paused. The trigger overrun flag is never set while in the software initiated single-scan mode.

The software initiated single-scan mode is useful in the following applications:

- Allows software complete control of the queue execution
- Allows the software to easily alternate between several queue sequences.

13.5.4.3.2 External Trigger Single-Scan Mode

The external trigger single-scan mode is available on both queue 1 and queue 2. The software programs the polarity of the external trigger edge that is to be detected, either a rising or a falling edge. The software must enable the scan to occur by setting the single-scan enable bit for the queue.

The first external trigger edge causes the queue to be executed one time. Each CCW is read and the indicated conversions are performed until an end-of-queue condition is encountered. After the queue is completed, the QADC64E clears the single-scan enable bit. Software may set the single-scan enable bit again to allow another scan of the queue to be initiated by the next external trigger edge.

The external trigger single-scan mode is useful when the input trigger rate can exceed the queue execution rate. Analog samples can be taken in sync with an external event, even though the software is not interested in data taken from every edge. The software can start the external trigger single-scan mode and get one set of data, and at a later time, start the queue again for the next set of samples.

When a pause bit is encountered during external trigger single-scan mode, another trigger event is required for queue execution to continue. Software involvement is not needed to enable queue execution to continue from the paused state.

13.5.4.3.3 External Gated Single-Scan Mode

The QADC64E provides external gating for queue 1 only. When external gated single-scan mode is selected, the input level on the associated external trigger signal enables and disables queue execution. The polarity of the external gated signal is fixed so only a high level opens the gate and a low level closes the gate. Once the gate is open, each CCW is read and the indicated conversions are performed until the gate is closed. Software must enable the scan to occur by setting the single-scan enable bit for queue 1. If a pause in a CCW is encountered, the pause flag will not set, and execution continues without pausing.

While the gate is open, queue 1 executes one time. Each CCW is read and the indicated conversions are performed until an end-of-queue condition is encountered. When queue 1 completes, the QADC64E sets the completion flag (CF1) and clears the single-scan enable bit. Software may set the single-scan enable bit again to allow another scan of queue 1 to be initiated during the next open gate.

If the gate closes before queue 1 completes execution, the current CCW completes, execution of queue 1 stops, the single-scan enable bit is cleared, and the PF1 bit is set. Software can read the CWPQ1 to

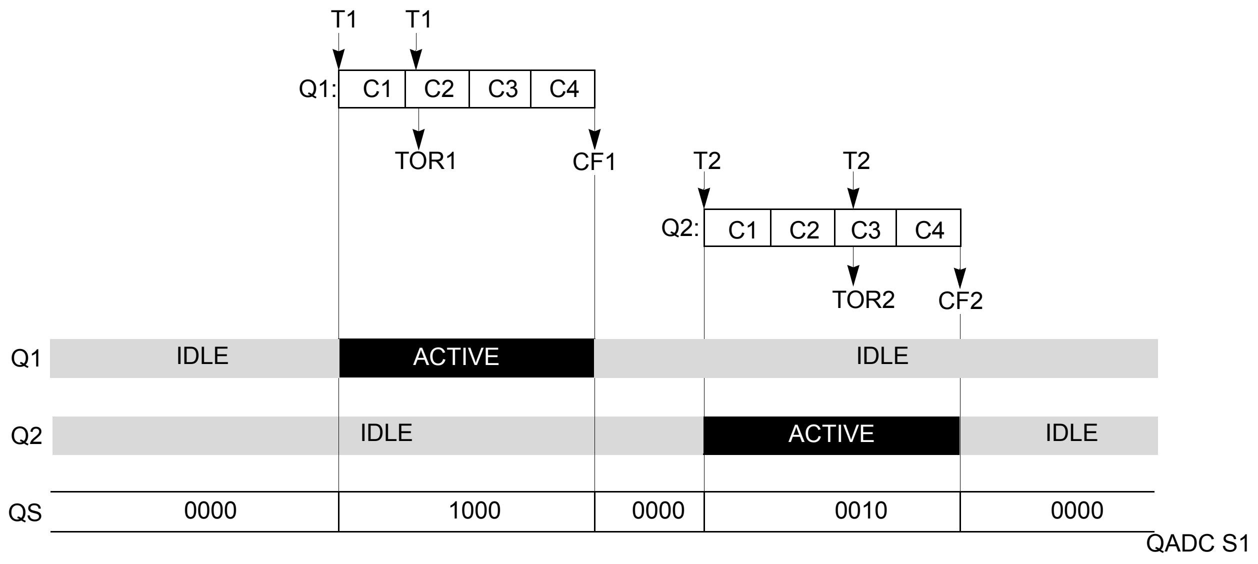


Figure 13-27. CCW Priority Situation 1

In situation S2 (Figure 13-27), more than one trigger event is recognized before servicing of a previous trigger event is complete, the trigger overrun bit is again set, but otherwise, the additional trigger events are ignored. After the queue is complete, the first newly detected trigger event causes queue execution to begin again. When the trigger event rate is high, a new trigger event can be seen very soon after completion of the previous queue, leaving software little time to retrieve the previous results. Also, when trigger events are occurring at a high rate for queue 1, the lower priority queue 2 channels may not get serviced at all.

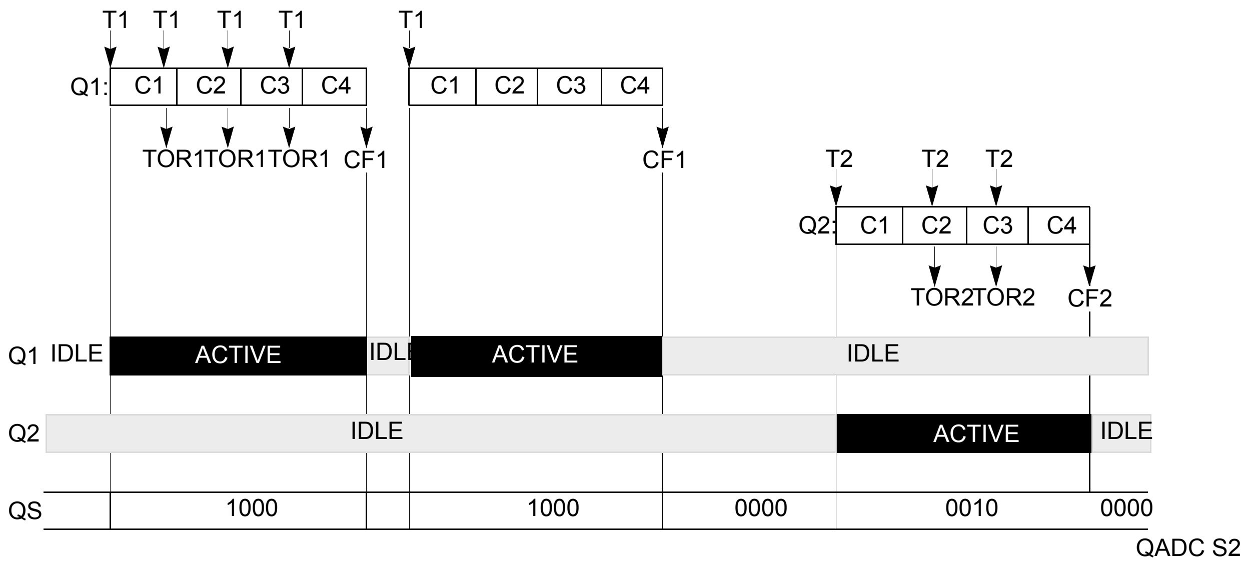


Figure 13-28. CCW Priority Situation 2

Situation S3 (Figure 13-28) shows that when the pause feature is in use, the trigger overrun error status bit is set the same way, and that queue execution continues unchanged.

The problem of how and when to combine digital and analog grounds arises from the large transients which the digital ground must handle. If the digital ground is not able to handle the large transients, the current from the large transients can return to ground through the analog ground. It is the excess current overflowing into the analog ground which causes performance degradation by developing a differential voltage between the true analog ground and the microcontroller's ground signal. The end result is that the ground observed by the analog circuit is no longer true ground and often ends in skewed results.

Two similar approaches designed to improve or eliminate the problems associated with grounding excess transient currents involve star-point ground systems. One approach is to star-point the different grounds at the power supply origin, thus keeping the ground isolated. Refer to Figure 13-51.

Another approach is to star-point the different grounds near the analog ground signal on the microcontroller by using small traces for connecting the non-analog grounds to the analog ground. The small traces are meant only to accommodate DC differences, not AC transients.

NOTE

This star-point scheme still requires adequate grounding for digital and analog subsystems in addition to the star-point ground.

Other suggestions for PCB layout in which the QADC64E is employed include:

- Analog ground must be low impedance to all analog ground points in the circuit.
- Bypass capacitors should be as close to the power signals as possible.

The analog ground should be isolated from the digital ground. This can be done by cutting a separate ground plane for the analog ground

- Non-minimum traces should be utilized for connecting bypass capacitors and filters to their corresponding ground/power points.
- Distance for trace runs should be minimized where possible

- The “on” resistance of the internal switches is 0 Ω and the “off” resistance is infinite

13.7.5.1 Analog Input Considerations

The source impedance of the analog signal to be measured and any intermediate filtering should be considered whether external multiplexing is used or not. Figure 13-53 shows the connection of eight typical analog signal sources to one QADC64E analog input signal through a separate multiplexer chip. Also, an example of an analog signal source connected directly to a QADC64E analog input channel is displayed.

attempts to write unimplemented data space, the QADC64E causes a bus error condition and no data is written.

- Attempts to read assignable data space in the unrestricted access mode when the space is programmed as supervisor space causes the bus master to assert a bus error condition and no data is returned.
- Attempts to write assignable data space in the unrestricted access mode when the space is programmed as supervisor space causes the bus master to assert a bus error condition and the register is not written.

Table 14-6. QADC64E Bus Error Response

S/U[1] Mode	SUPV Bit	Supervisor-Only Register	Supervisor/ Unrestricted Register	Reserved/ Unimplemented Register
U	0	QADC64E bus error[2]	Valid access[4]	QADC64E bus error[2]
U	1	Master bus error[3]	Master bus error[3]	Master bus error[3]
S	0	Valid access	Valid access	QADC64E bus error[2]
S	1	Valid access	Valid access	QADC64E bus error[2]

1 S/U = Supervisor/Unrestricted

2 QADC64E bus error = Caused by QADC64E

3 Master bus error = Caused by bus master

4 Access to QADCTEST register will act as a reserved/unimplemented register unless in factory test mode

The bus master indicates the supervisor and user space access with the function code bits (FC[2:0]) on the IMB3. For privilege violations, refer to Chapter 9, "External Bus Interface" to determine the consequence of a bus error cycle termination.

The supervisor-only data space segment contains the QADC64E global registers, which include the QADCMCR, QADCINT and QADCTEST. The supervisor/unrestricted space designation for the CCW table, the result word table and the remaining QADC64E registers is programmable.

14.3.2 QADC64E Interrupt Register

QADCINT specifies the priority level of QADC64E interrupt requests. The interrupt level for queue 1 and queue 2 may be different. The interrupt register is read/write accessible in supervisor data space only. The implemented interrupt register fields can be read and written, reserved bits read zero and writes have no effect. They are typically written once when the software initializes the QADC64E, and not changed afterwards.

Table 17-5. MIOS14MCR Bit Descriptions (continued)

Bits	Name	Description
8	SUPV	Supervisor data space selector — The SUPV bit tells if the address space from 0x30 6000 to 0x30 67FF in the MIOS14 is accessed at the supervisor privilege level (See Figure 17-2). When cleared, these addresses are accessed at the unrestricted privilege level. The SUPV bit is cleared by reset. 0 Unrestricted Data Space. 1 Supervisor Data Space.
9:15	—	Reserved. These bits are used for the IARB (interrupt arbitration ID) field in MIOS14 implementations that use hardware interrupt arbitration. These bits are not used on MPC561/MPC563.

17.7 MIOS14 Counter Prescaler Submodule (MCPSM)

The MIOS14 counter prescaler submodule (MCPSM) divides the MIOS14 clock (f_{SYS}) to generate the counter clock. It is designed to provide all the submodules with the same division of the main MIOS14 clock (division of f_{SYS}). It uses a 4-bit modulus counter. The clock signal is prescaled by loading the value of the clock prescaler register into the prescaler counter every time it overflows. This allows all prescaling factors between 2 and 16. Counting is enabled by asserting MCPSMSCR[PREN]. The counter can be stopped at any time by negating this bit, thereby stopping all submodules using the output of the MCPSM (counter clock). A block diagram of the MCPSM is given in Figure 17-8.

The following sections describe the MCPSM in detail.

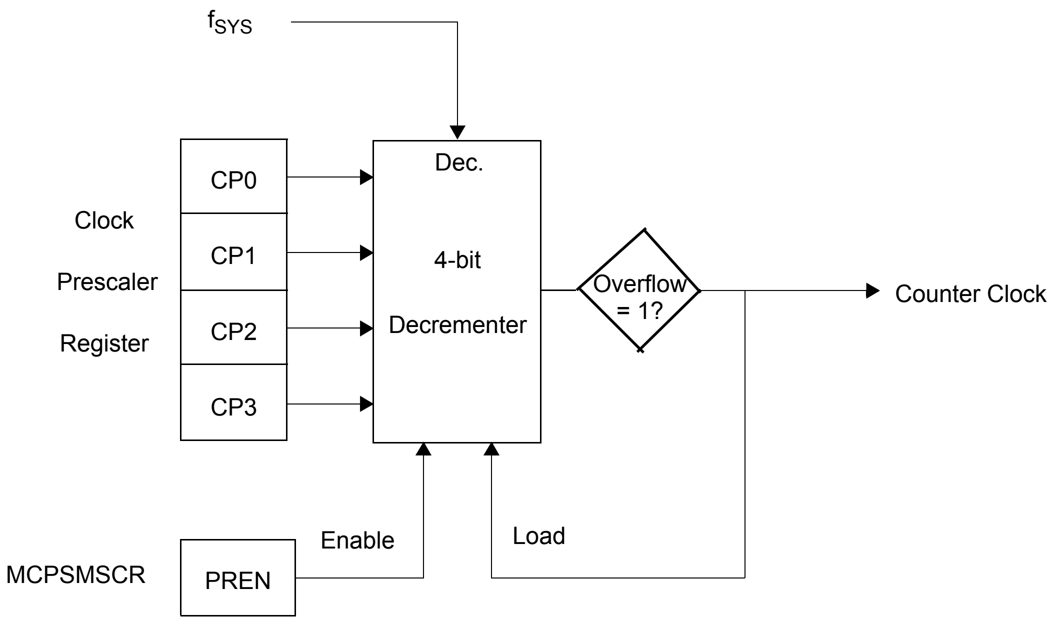


Figure 17-8. MCPSM Block Diagram

17.7.1 MCPSM Features

- Centralized counter clock generator

19.4.10 Channel Interrupt Status Register (CISR)

The channel interrupt status register (CISR) contains one interrupt status flag per channel. Time functions specify via microcode when an interrupt flag is set. Setting a flag causes the TPU3 to make an interrupt service request if the corresponding CIER bit is set. To clear a status flag, read CISR, then write a zero to the appropriate bit.

NOTE

CISR is the only TPU3 register that can be accessed on a byte basis.

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	LSB 15
Field	CH 15	CH 14	CH 13	CH 12	CH 11	CH 10	CH 9	CH 8	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0
$\overline{\text{SRESET}}$	0000_0000_0000_0000															
Addr	0x30 4020 (TPU_A), 0x30 4420 (TPU_B)															

Figure 19-20. CISR — Channel Interrupt Status Register

Table 19-17. CISR Bit Descriptions

Bits	Name	Description
0:15	CH[15:0]	Channel interrupt status 0 Channel interrupt not asserted 1 Channel interrupt asserted

19.4.11 TPU3 Module Configuration Register 2 (TPUMCR2)

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	LSB 15
Field	—							DIV2	SOFTRST	ETBANK		FPSCK			T2CF	DTPU
$\overline{\text{SRESET}}$	0000_0000_0000_0000															
Addr	0x30 4028 (TPU_A), 0x30 4428 (TPU_B)															

Figure 19-21. TPUMCR2 — TPU Module Configuration Register 2

Table 19-18. TPUMCR2 Bit Descriptions

Bits	Name	Description
0:6	—	Reserved
7	DIV2	Divide by 2 control. When asserted, the DIV2 bit, along with the TCR1P bit and the PSCK bit in the TPUMCR, determines the rate of the TCR1 counter in the TPU3. If set, the TCR1 counter increments at a rate of two system clocks. If negated, TCR1 increments at the rate determined by control bits in the TCR1P and PSCK fields. 0 TCR1 increments at rate determined by control bits in the TCR1P and PSCK fields of the TPUMCR register 1 Causes TCR1 counter to increment at a rate of the system clock divided by two

19.4.15 TPU3 Parameter RAM

The channel parameter registers are organized as one hundred 16-bit words of RAM. Channels 0 to 15 have eight parameters. The parameter registers constitute a shared work space for communication between the CPU and the TPU3. The TPU3 can only access data in the parameter RAM. Refer to Table 19-24.

Table 19-24. Parameter RAM Address Offset Map

Channel Number	Parameter							
	0	1	2	3	4	5	6	7
0	0x30 4100(A) 0x30 4500(B))	0x30 4102A(A) 0x30 4502(B)	0x30 4104(A) 0x30 4504(B)	0x30 4106(A) 0x30 4506(B)	0x30 4108(A) 0x30 4508(B)	0x30 410A(A) 0x30 450A(B)	0x30 410C(A) 0x30 450C(B)	0x30 410E(A) 0x30 450E(B)
1	0x30 4110(A) 0x30 4510(B)	0x30 4112(A) 0x30 4512(B)	0x30 4114(A) 0x30 4514(B)	0x30 4116(A) 0x30 4516(B)	0x30 4118(A) 0x30 4518(B)	0x30 411A(A) 0x30 451A(B)	0x30 411C(A) 0x30 451C(B)	0x30 411E(A) 0x30 451E(B)
2	0x30 4120(A) 0x30 4520(B)	0x30 4122(A) 0x30 4522(B)	0x30 4124(A) 0x30 4524(B)	0x30 4126(A) 0x30 4526(B)	0x30 4128(A) 0x30 4528(B)	0x30 412A(A) 0x30 452A(B)	0x30 412C(A) 0x30 452C(B))	0x30 412E(A) 0x30 452E(B)
3	0x30 4130(A) 0x30 4530(B)	0x30 4132(A) 0x30 4532(B)	0x30 4134(A) 0x30 4534(B)	0x30 4136(A) 0x30 4536(B)	0x30 4138(A) 0x30 4538(B)	0x30 413A(A) 0x30 453A(B)	0x30 413C(A) 0x30 453C(B)	0x30 413E(A) 0x30 453E(B)
4	0x30 4140(A) 0x30 4540(B)	0x30 4142(A) 0x30 4542(B)	0x30 4144(A) 0x30 4544(B)	0x30 4146(A) 0x30 4546(B)	0x30 4148(A) 0x30 4548(B)	0x30 414A(A) 0x30 454A(B)	0x30 414C(A) 0x30 454C(B)	0x30 414E(A) 0x30 454E(B)
5	0x30 4150(A) 0x30 4550(B)	0x30 4152(A) 0x30 4552(B)	0x30 4154(A) 0x30 4554(B)	0x30 4156(A) 0x30 4556(B)	0x30 4158(A) 0x30 4558(B)	0x30 415A(A) 0x30 455A(B)	0x30 415C(A) 0x30 455C(B)	0x30 415E(A) 0x30 455E(B)
6	0x30 4160(A) 0x30 4560(B)	0x30 4162(A) 0x30 4562(B)	0x30 4164(A) 0x30 4564(B)	0x30 4166(A) 0x30 4566(B)	0x30 4168(A) 0x30 4568(B)	0x30 416A(A) 0x30 456A(B)	0x30 416C(A) 0x30 456C(B)	0x30 416E(A) 0x30 456E(B)
7	0x30 4170(A) 0x30 4570(B)	0x30 4172(A) 0x30 4572(B)	0x30 4174(A) 0x30 4574(B)	0x30 4176(A) 0x30 4576(B)	0x30 4178(A) 0x30 4578(B)	0x30 417A(A) 0x30 457A(B)	0x30 417C(A) 0x30 457C(B)	0x30 417E(A) 0x30 457E(B)
8	0x30 4180(A) 0x30 4580(B)	0x30 4182(A) 0x30 4582(B)	0x30 4184(A) 0x30 4585(B)	0x30 4186(A) 0x30 4586(B)	0x30 4188(A) 0x30 4588(B)	0x30 418A(A) 0x30 458A(B)	0x30 418C(A) 0x30 458C(B)	0x30 418E(A) 0x30 458E(B)
9	0x30 4190(A) 0x30 4590(B)	0x30 4192(A) 0x30 4592(B)	0x30 4194(A) 0x30 4594(B)	0x30 4196(A) 0x30 4596(B)	0x30 4198(A) 0x30 4598(B)	0x30 419A(A) 0x30 459A(B)	0x30 419C(A) 0x30 459C(B)	0x30 419E(A) 0x30 459E(B)
10	0x30 41A0(A) 0x30 45A0(B)	0x30 41A2(A) 0x30 45A2(B)	0x30 41A4(A) 0x30 45A4(B)	0x30 41A6(A) 0x30 45A6(B)	0x30 41A8(A) 0x30 45A8(B)	0x30 41AA(A) 0x30 45AA(B)	0x30 41AC(A) 0x30 45AC(B)	0x30 41AE(A) 0x30 45AE(B)
11	0x30 41B0(A) 0x30 45B0(B)	0x30 41B2(A) 0x30 45B2(B)	0x30 41B4(A) 0x30 45B4(B)	0x30 41B6(A) 0x30 45B6(B)	0x30 41B8(A) 0x30 45B8(B)	0x30 41BA(A) 0x30 45BA(B)	0x30 41BC(A) 0x30 45BC(B)	0x30 41BE(A) 0x30 45BE(B)
12	0x30 41C0(A) 0x30 45C0(B)	0x30 41C2(A) 0x30 45C2(B)	0x30 41C4(A) 0x30 45C4(B)	0x30 41C6(A) 0x30 45C6(B)	0x30 41C8(A) 0x30 45C8(B)	0x30 41CA(A) 0x30 45CA(B)	0x30 41CC(A) 0x30 45CC(B)	0x30 41CE(A) 0x30 45CE(B)
13	0x30 41D0(A) 0x30 45D0(B)	0x30 41D2(A) 0x30 45D2(B)	0x30 41D4(A) 0x30 45D4(B)	0x30 41D6(A) 0x30 45D6(B)	0x30 41D8(A) 0x30 45D8(B)	0x30 41DA(A) 0x30 45DA(B)	0x30 41DC(A) 0x30 45DC(B)	0x30 41DE(A) 0x30 45DE(B)
14	0x30 41E0(A) 0x30 45E0(B)	0x30 41E2(A) 0x30 45E2(B)	0x30 41E4(A) 0x30 45E4(B)	0x30 41E6(A) 0x30 45E6(B)	0x30 41E8(A) 0x30 45E8(B)	0x30 41EA(A) 0x30 45EA(B)	0x30 41EC(A) 0x30 45EC(B)	0x30 41EE(A) 0x30 45EE(B)
15	0x30 41F0(A) 0x30 45F0(B)	0x30 41F2(A) 0x30 45F2(B)	0x30 41F4(A) 0x30 45F4(B)	0x30 41F6(A) 0x30 45F6(B)	0x30 41F8(A) 0x30 45F8(B)	0x30 41FA(A) 0x30 45FA(B)	0x30 41FC(A) 0x30 45FC(B)	0x30 41FE(A) 0x30 45FE(B)

19.5 Time Functions

Descriptions of the MPC561/MPC563 pre-programmed time functions are shown in Appendix D, “TPU3 ROM Functions.”

D.15 Output Compare (OC)

The output compare (OC) function generates a rising edge, falling edge, or a toggle of the previous edge: immediately upon RCPU initiation (generating a pulse with a length equal to a programmable delay time), after a programmable delay time, or continuously. Upon receiving a link from a channel, OC references, without RCPU interaction, a specifiable period and calculates an offset that is equal to the period x the ratio, where the ratio is a supplied parameter.

This algorithm generates, with each high/low time, a 50% duty-cycle continuous square equal to the calculated offset. Due to offset calculation, there is an initial link time before continuous pulse generation begins. See Freescale TPU Progamming Note *Output Compare TPU Function (OC), (TPUPN12/D).*

Figure D-26 shows the host interface areas and parameter RAM for the OC function.

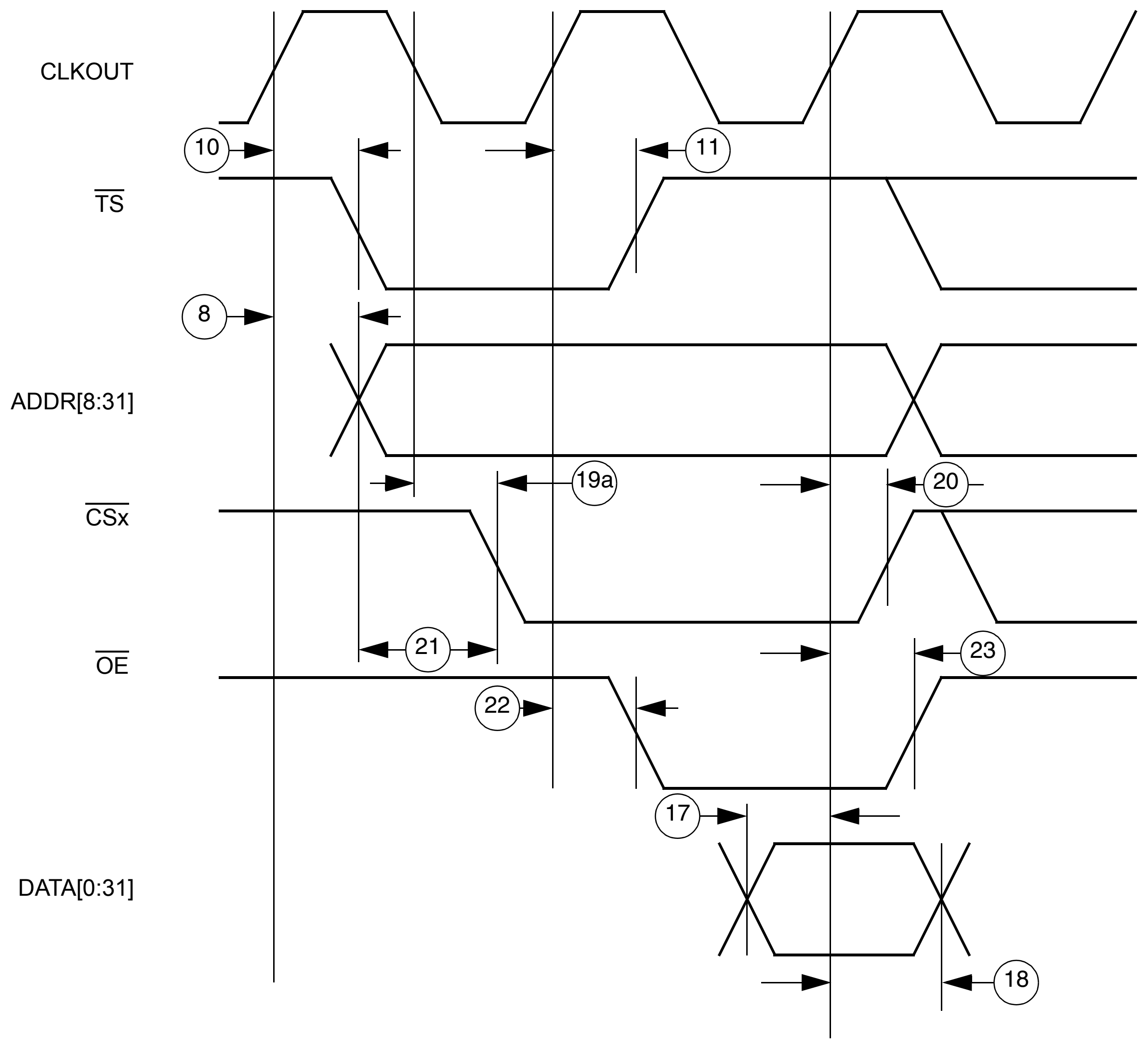


Figure F-17. External Bus Read Timing (GPCM Controlled – TRLX = ‘0’ ACS = ‘10’)

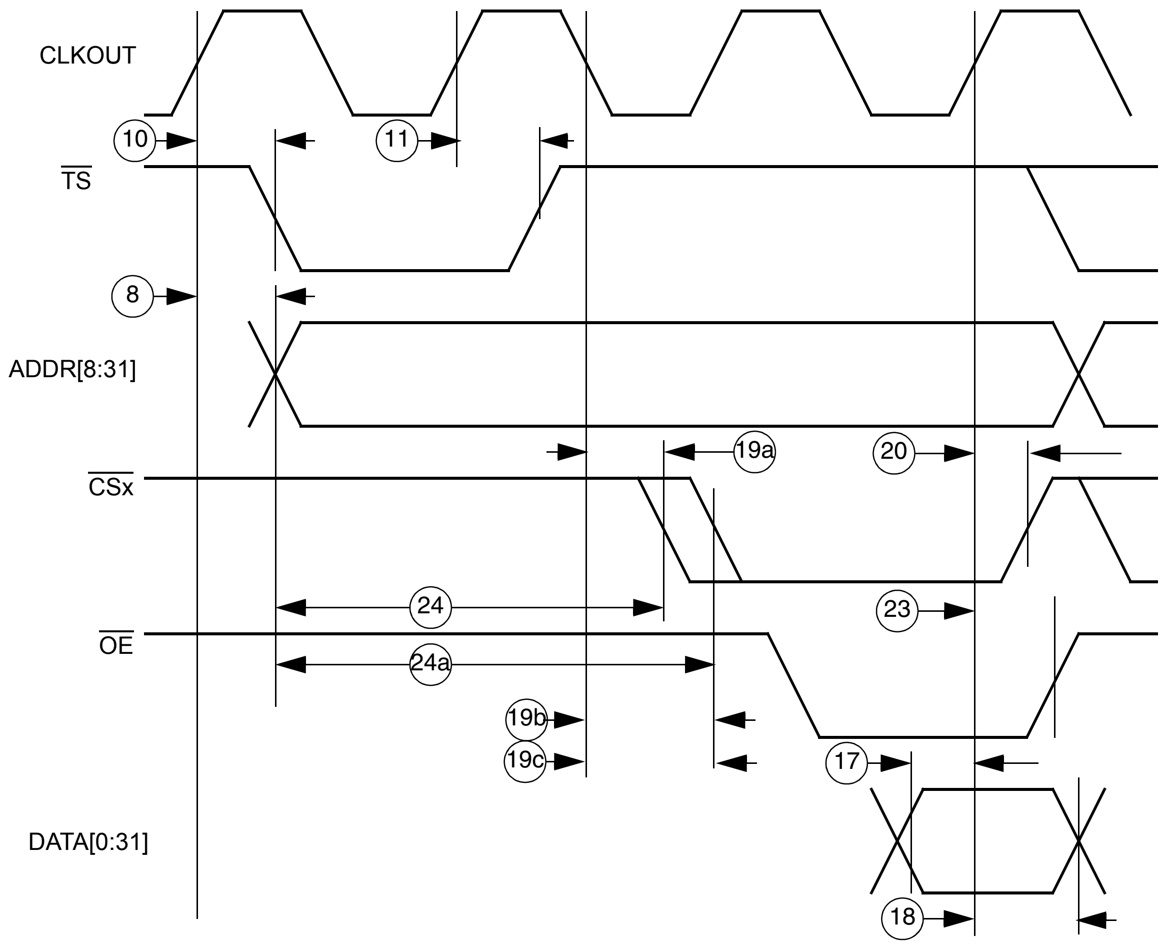


Figure F-19. External Bus Read Timing (GPCM Controlled – TRLX = ‘1’, ACS = ‘10’, ACS = ‘11’)

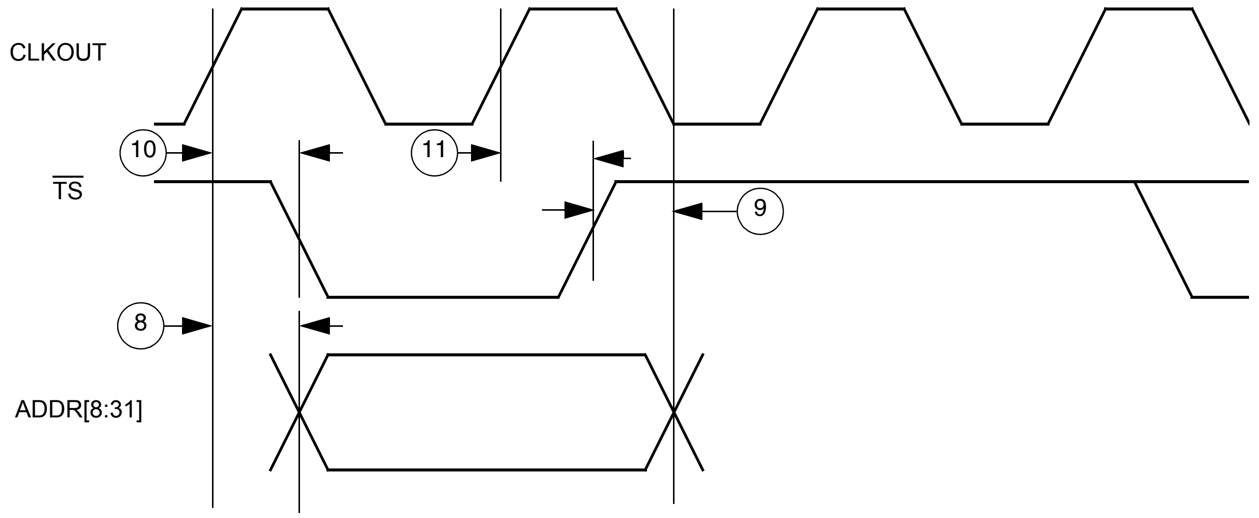


Figure F-20. Address Show Cycle Bus Timing

[1] JTAG timing (TCK) is only tested at 10 MHz. TCK is the operating clock of the MPC561/MPC563 in JTAG mode.

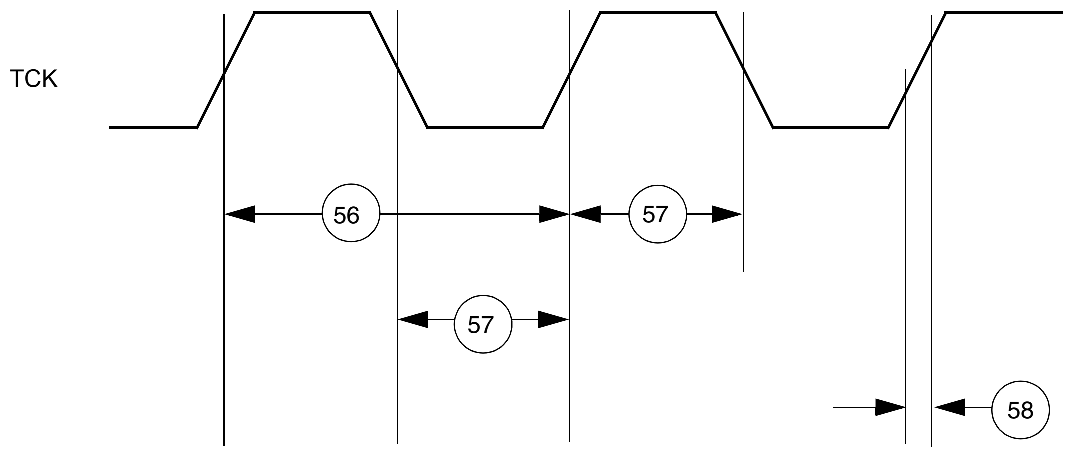


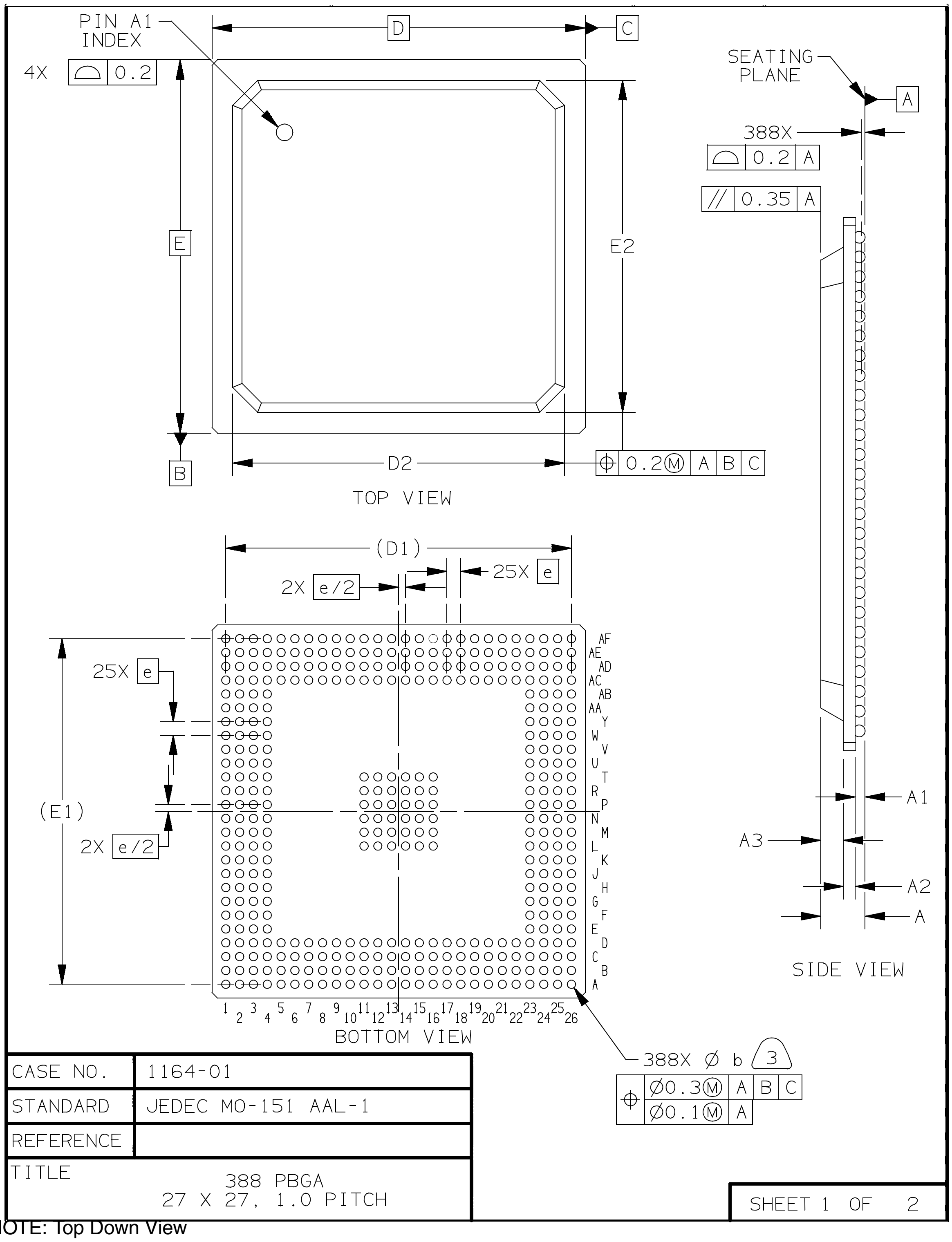
Figure F-37. JTAG Test Clock Input Timing

Table F-28. MPC561/MPC563 Signal Names and Pin Names (continued)

Signal Name	Pin Name	Ball Assignment
$\overline{\text{BR}}$/VF1/IWP2	br_b_vf1_iwp2	R4
$\overline{\text{BB}}$/VF2/IWP3	bb_b_vf2_iwp3	R2
IWP[0:1]/VFLS[0:1]	iwp0_vfls0	N2
	iwp1_vfls1	N3
TMS/$\overline{\text{EVTI}}$	tms_evti_b	M2
TDI/DSDI/MDI0	tdi_dsdi_mdi0	M1
TCK/DSCK/MCKI	tck_dsck_mcki	L2
TDO/DSDO/MDO0	tdo_dsdo_mdo0	M4
JCOMP/$\overline{\text{RSTI}}$	jcomp_rsti_b	L1
XTAL	xtal	AD26
EXTAL	extal	AC26
XFC	xfc	AA26
CLKOUT	clkout	U23
EXTCLK	extclk	V24
ENGCLK/BUCLK	engclk_buclk	V26
QSMCM		
PCS0/$\overline{\text{SS}}$/QGPIO0	pcs0_ss_b_qgpio0	N25
PCS[1:3]/QGPIO[1:3]	pcs1_qgpio1	N24
	pcs2_qgpio2	N23
	pcs3_qgpio3	P26
$\overline{\text{MISO}}$/QGPIO4	miso_b_qgpio4	P25
$\overline{\text{MOSI}}$/QGPIO5	mosi_b_qgpio5	P24
SCK/QGPIO6	sck_qgpio6	P23
TXD1/QGPO1	txd1_qgpo1	R25
TXD2/QGPO2/C_CNTX0	txd2_qgpo2_c_cntx0	R24
RXD1/QGPI1	rxd1_qgpi1	R23
RXD2/QGPI2/C_CNRX0	rxd2_qgpi2_c_cnrx0	T26
MIOS14		

Table F-28. MPC561/MPC563 Signal Names and Pin Names (continued)

Signal Name	Pin Name	Ball Assignment
MDA[11:15]	mda11	C20
	mda12	D20
	mda13	A21
	mda14	B21
	mda15	C21
MDA[27:31]	mda27	D21
	mda28	A22
	mda29	B22
	mda30	F24
	mda31	F25
MPWM[0:1]/MDI[1:2]	mpwm0_mdi1	F26
	mpwm1_mdo2	G23
MPWM2/PPM_TX1	mpwm2_ppm_tx1	G26
MPWM3/PPM_RX1	mpwm3_ppm_rx1	G25
MPWM16	mpwm16	G24
MPWM17/MDO3	mpwm17_mdo3	H23
MPWM[18:19]/MDO[6:7]	mpwm18_mdo6	H24
	mpwm19_mdo7	H25
VF0/MPIO32B0/MDO1	vf0_mpio32b0_mdo1	L23
VF1/MPIO32B1/MCKO	vf1_mpio32b1_mcko	L24
VF2/MPIO32B2/$\overline{\text{MSEI}}$	vf2_mpio32b2_msei_b	M24
VFLS0/MPIO32B3/$\overline{\text{MSEO}}$	vfls0_mpio32b3_mseo_b	M25
VFLS1/MPIO32B4	vfls1_mpio32b4	M26
MPIO32B5/MDO5	mpio32b5_mdo5	H26
MPIO32B6/MPWM4/MDO6	mpio32b6_mpwm4_mdo6	J23
MPIO32B7/MPWM5	mpio32b7_mpwm5	J24
MPIO32B[8:9]/MPWM[20:21]	mpio32b8_mpwm20	J25
	mpio32b9_mpwm21	J26
MPIO32B10/PPM_TSYNC	mpio32b10_ppm_tsync	K25
MPIO32B11/C_CNRX0	mpio32b11_c_cnrx0	K24
MPIO32B12/C_CNTX0	mpio32b12_c_cntx0	K23
MPIO32B13/PPM_TCLK	mpio32b13_ppm_tclk	K26
MPIO32B14/PPM_RX0	mpio32b14_ppm_rx0	L26



CASE NO.	1164-01
STANDARD	JEDEC MO-151 AAL-1
REFERENCE	
TITLE	388 PBGA 27 X 27, 1.0 PITCH

1 NOTE: Top Down View

Figure F-64. MPC561/MPC563 Package Footprint (1 of 2)