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	Coldfire V3
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	EBI/EMI, I²C, UART/USART
Peripherals	DMA, POR, WDT
Number of I/O	16
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	208-BFQFP
Supplier Device Package	208-FQFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcf5307ft66b

Terminology and Notational Conventions

Table ii shows notational conventions used throughout this document.

Table ii Notational Conventions

Instruction	Operand Syntax
Opcode Wildcard	
cc	Logical condition (example: NE for not equal)
Register Specifications	
An	Any address register n (example: A3 is address register 3)
Ay,Ax	Source and destination address registers, respectively
Dn	Any data register n (example: D5 is data register 5)
Dy,Dx	Source and destination data registers, respectively
Rc	Any control register (example VBR is the vector base register)
Rm	MAC registers (ACC, MAC, MASK)
Rn	Any address or data register
Rw	Destination register w (used for MAC instructions only)
Ry,Rx	Any source and destination registers, respectively
Xi	index register i (can be an address or data register: Ai, Di)
Register Names	
ACC	MAC accumulator register
CCR	Condition code register (lower byte of SR)
MACSR	MAC status register
MASK	MAC mask register
PC	Program counter
SR	Status register
Port Name	
PSTDDATA	Processor status/debug data port
Miscellaneous Operands	
#<data>	Immediate data following the 16-bit operation word of the instruction
Í	Effective address
<ea>y,<ea>x	Source and destination effective addresses, respectively
<label>	Assembly language program label
<list>	List of registers for MOVEM instruction (example: D3–D0)
<shift>	Shift operation: shift left (<<), shift right (>>)
<size>	Operand data size: byte (B), word (W), longword (L)
bc	Both instruction and data caches
dc	Data cache

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Features common to many embedded applications, such as DMAs, various DRAM controller interfaces, and on-chip memories, are integrated using advanced process technologies.

The MCF5307 extends the legacy of Motorola's 68K family by providing a compatible path for 68K and ColdFire customers in which development tools and customer code can be leveraged. In fact, customers moving from 68K to ColdFire can use code translation and emulation tools that facilitate modifying 68K assembly code to the ColdFire architecture.

Based on the concept of variable-length RISC technology, the ColdFire family combines the architectural simplicity of conventional 32-bit RISC with a memory-saving, variable-length instruction set. In defining the ColdFire architecture for embedded processing applications, a 68K-code compatible core combines performance advantages of a RISC architecture with the optimum code density of a streamlined, variable-length M68000 instruction set.

By using a variable-length instruction set architecture, embedded system designers using ColdFire RISC processors enjoy significant advantages over conventional fixed-length RISC architectures. The denser binary code for ColdFire processors consumes less memory than many fixed-length instruction set RISC processors available. This improved code density means more efficient system memory use for a given application and allows use of slower, less costly memory to help achieve a target performance level.

The MCF5307 is the first standard product to implement the Version 3 ColdFire microprocessor core. To reach higher levels of frequency and performance, numerous enhancements were made to the V2 architecture. Most notable are a deeper instruction pipeline, branch acceleration, and a unified cache, which together provide 75 (Dhrystone 2.1) MIPS at 90 MHz. Increasing the internal speed of the core also allows higher performance while providing the system designer with an easy-to-use lower speed system interface. The processor complex frequency is an integer multiple, 2 to 4 times, of the external bus frequency. The core clock can be stopped to support a low-power mode.

Serial communication channels are provided by an I^2C interface module and two programmable full-duplex UARTs. Four channels of DMA allow for fast data transfer using a programmable burst mode independent of processor execution. The two 16-bit general-purpose multimode timers provide separate input and output signals. For system protection, the processor includes a programmable 16-bit software watchdog timer. In addition, common system functions such as chip selects, interrupt control, bus arbitration, and an IEEE 1149.1 JTAG module are included. A sophisticated debug interface supports background-debug mode plus real-time trace and debug with expanded flexibility of on-chip breakpoint registers. This interface is present in all ColdFire standard products and allows common emulator support across the entire family of microprocessors.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

On-chip breakpoint resources include the following:

- Configuration/status register (CSR)
- Background debug mode (BDM) address attributes register (BAAR)
- Bus attributes and mask register (AATR)
- Breakpoint registers. These can be used to define triggers combining address, data, and PC conditions in single- or dual-level definitions. They include the following:
  - PC breakpoint register (PBR)
  - PC breakpoint mask register (PBMR)
  - Data operand address breakpoint registers (ABHR/ABLR)
  - Data breakpoint register (DBR)
- Data breakpoint mask register (DBMR)
- Trigger definition register (TDR) can be programmed to generate a processor halt or initiate a debug interrupt exception.

These registers can be accessed through the dedicated debug serial communication channel, or from the processor's supervisor programming model, using the WDEBUG instruction.

The enhancements of the Revision B debug specification are fully backward-compatible with the A revision. For more information, see Chapter 5, "Debug Support."

2.2 Programming Model

The MCF5307 programming model consists of three instruction and register groups—user, MAC (also user-mode), and supervisor, shown in Figure 2-2. User mode programs are restricted to user and MAC instructions and programming models. Supervisor-mode system software can reference all user-mode and MAC instructions and registers and additional supervisor instructions and control registers. The user or supervisor programming model is selected based on SR[S]. The following sections describe the registers in the user, MAC, and supervisor programming models.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 2-6. Notational Conventions (Continued)

Instruction	Operand Syntax
P	Branch prediction
C	Carry
N	Negative
V	Overflow
X	Extend
Z	Zero

2.6.1 Instruction Set Summary

Table 2-7 lists implemented user-mode instructions by opcode.

Table 2-7. User-Mode Instruction Set Summary

Instruction	Operand Syntax	Operand Size	Operation
ADD	Dy,<ea>x <ea>y,Dx	.L .L	Source + destination → destination
ADDA	<ea>y,Ax	.L	Source + destination → destination
ADDI	#<data>,Dx	.L	Immediate data + destination → destination
ADDQ	#<data>,<ea>x	.L	Immediate data + destination → destination
ADDX	Dy,Dx	.L	Source + destination + X → destination
AND	Dy,<ea>x <ea>y,Dx	.L .L	Source & destination → destination
ANDI	#<data>,Dx	.L	Immediate data & destination → destination
ASL	Dy,Dx #<data>,Dx	.L .L	X/C ← (Dx << Dy) ← 0 X/C ← (Dx << #<data>) ← 0
ASR	Dy,Dx #<data>,Dx	.L .L	MSB → (Dx >> Dy) → X/C MSB → (Dx >> #<data>) → X/C
Bcc	<label>	.B,.W	If condition true, then PC + 2 + d_n → PC
BCHG	Dy,<ea>x #<data>,<ea-1>x	.B,.L .B,.L	~(<bit number> of destination) → Z, Bit of destination
BCLR	Dy,<ea>x #<data>,<ea-1>x	.B,.L .B,.L	~(<bit number> of destination) → Z; 0 → bit of destination
BRA	<label>	.B,.W	PC + 2 + d_n → PC
BSET	Dy,<ea>x #<data>,<ea-1>x	.B,.L .B,.L	~(<bit number> of destination) → Z; 1→ bit of destination
BSR	<label>	.B,.W	SP – 4 → SP; next sequential PC→ (SP); PC + 2 + d_n → PC
BTST	Dy,<ea>x #<data>,<ea-1>x	.B,.L .B,.L	~(<bit number> of destination) → Z
CLR	<ea>y,Dx	.B,.W,.L	0 → destination
CMP	<ea>y,Ax	.L	Destination – source
CMPA	<ea>y,Dx	.L	Destination – source

For More Information On This Product,
Go to: www.freescale.com
Freescale Semiconductor, Inc.

These registers are described as follows:

- Accumulator (ACC)—This 32-bit, read/write, general-purpose register is used to accumulate the results of MAC operations.
- Mask register (MASK)—This 16-bit general-purpose register provides an optional address mask for MAC instructions that fetch operands from memory. It is useful in the implementation of circular queues in operand memory.
- MAC status register (MACSR)—This 8-bit register defines configuration of the MAC unit and contains indicator flags affected by MAC instructions. Unless noted otherwise, the setting of MACSR indicator flags is based on the final result, that is, the result of the final operation involving the product and accumulator.

3.1.2 General Operation

The MAC unit supports the ColdFire integer multiply instructions (MULS and MULU) and provides additional functionality for multiply-accumulate operations. The added MAC instructions to the ColdFire ISA provide for the multiplication of two numbers, followed by the addition or subtraction of this number to or from the value contained in the accumulator. The product may be optionally shifted left or right one bit before the addition or subtraction takes place. Hardware support for saturation arithmetic may be enabled to minimize software overhead when dealing with potential overflow conditions using signed or unsigned operands.

These MAC operations treat the operands as one of the following formats:

- Signed integers
- Unsigned integers
- Signed, fixed-point, fractional numbers

To maintain compactness, the MAC module is optimized for 16-bit multiplications. Two 16-bit operands produce a 32-bit product. Longword operations are performed by reusing the 16-bit multiplier array at the expense of a small amount of extra control logic. Again, the product of two 32-bit operands is a 32-bit result. For longword integer operations, only the least significant 32 bits of the product are calculated. For fractional operations, the entire 63-bit product is calculated and then either truncated or rounded to a 32-bit result using the round-to-nearest (even) method.

Because the multiplier array is implemented in a 3-stage pipeline, MAC instructions can have an effective issue rate of one clock for word operations, three for longword integer operations, and four for 32-bit fractional operations. Arithmetic operations use register-based input operands, and summed values are stored internally in the accumulator. Thus, an additional MOVE instruction is necessary to store data in a general-purpose register. MAC instructions can choose the upper or lower word of a register as the input, which helps filtering operations in which one data register is loaded with input data and another is loaded with coefficient data. Two 16-bit MAC operations can be performed without fetching additional operands between instructions by alternating the word choice

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 5-7. BAAR Field Descriptions

Bits	Name	Description
7	R	Read/write 0 Write 1 Read
6–5	SZ	Size 00 Longword 01 Byte 10 Word 11 Reserved
4–3	TT	Transfer type. See the TT definition in Table 5-4.
2–0	TM	Transfer modifier. See the TM definition in Table 5-4.

5.4.4 Configuration/Status Register (CSR)

The configuration/status register (CSR) defines the debug configuration for the processor and memory subsystem and contains status information from the breakpoint logic.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	BSTAT				FOF	TRG	HALT	BKPT	HRL				—	BKD	—	IPW
Reset	0000				0	0	0	0	0001				—	—	—	0
R/W[1]	R				R	R	R	R	R				—	—	—	R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	MAP	TRC	EMU	DDC		UHE	BTB		—[2]	NPL	IPI	SSM	—			
Reset	0	0	0	00		0	00		0	0	—	0	—			
R/W	R/W	R/W	R/W	R/W		R/W	R/W		R	R/W	—	R/W	—			
DRc[4–0]	0x00															

1 CSR is write-only from the programming model. It can be read from and written to through the BDM port. CSR is accessible in supervisor mode as debug control register 0x00 using the WDEBUG instruction and through the BDM port using the RDMREG and WDMREG commands.

2 Bit 7 is reserved for Motorola use and must be written as a zero.

Figure 5-8. Configuration/Status Register (CSR)

Table 5-8 describes CSR fields.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

5.5.3.3.5 Dump Memory Block (DUMP)

DUMP is used with the READ command to access large blocks of memory. An initial READ is executed to set up the starting address of the block and to retrieve the first result. If an initial READ is not executed before the first DUMP, an illegal command response is returned. The DUMP command retrieves subsequent operands. The initial address is incremented by the operand size (1, 2, or 4) and saved in a temporary register. Subsequent DUMP commands use this address, perform the memory read, increment it by the current operand size, and store the updated address in the temporary register.

NOTE:

DUMP does not check for a valid address; it is a valid command only when preceded by NOP, READ, or another DUMP command. Otherwise, an illegal command response is returned. NOP can be used for intercommand padding without corrupting the address pointer.

The size field is examined each time a DUMP command is processed, allowing the operand size to be dynamically altered.

Command/Result Formats:

		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte	Command	0x1				0xD				0x0				0x0			
	Result	X	X	X	X	X	X	X	X	D[7:0]							
Word	Command	0x1				0xD				0x4				0x0			
	Result	D[15:0]															
Longword	Command	0x1				0xD				0x8				0x0			
	Result	D[31:16]															
		D[15:0]															

Figure 5-26. DUMP Command/Result Formats

Freescale Semiconductor, Inc.

- Control logic and state machine—Generates all DRAM signals, taking bus cycle characteristic data from the block logic, along with hit information to generate DRAM accesses. Handles refresh requests from the refresh counter.
  - DRAM control register (DCR)—Contains data to control refresh operation of the DRAM controller. Both memory blocks are refreshed concurrently as controlled by DCR[RC].
  - Refresh counter—Determines when refresh should occur, determined by the value of DCR[RC]. It generates a refresh request to the control block.
- Hit logic—Compares address and attribute signals of a current DRAM bus cycle to both DACRs to determine if a DRAM block is being accessed. Hits are passed to the control logic along with characteristics of the bus cycle to be generated.
- Page hit logic—Determines if the next DRAM access is in the same DRAM page as the previous one. This information is passed on to the control logic.
- Address multiplexing—Multiplexes addresses to allow column and row addresses to share pins. This allows glueless interface to DRAMs.

11.2 DRAM Controller Operation

The DRAM controller mode is programmed through DCR[SO]. Asynchronous mode (SO = 0) includes support for page mode and EDO DRAMs. Synchronous mode is designed to work with industry-standard SDRAMs. These modes act very differently from one another, especially regarding the use of DRAM registers and pins. Memory blocks cannot operate in different modes; both are either synchronous or asynchronous.

11.2.1 DRAM Controller Registers

The DRAM controller registers memory map, Table 11-1, is the same regardless of whether asynchronous or synchronous DRAM is used, although bit configurations may vary.

Table 11-1. DRAM Controller Registers

MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x100	DRAM control register (DCR) [p. 11-4]		Reserved	
0x104	Reserved			
0x108	DRAM address and control register 0 (DACR0) [p. 11-5]			
0x10C	DRAM mask register block 0 (DMR0) [p. 11-7]			
0x110	DRAM address and control register 1 (DACR1) [p. 11-5]			
0x114	DRAM mask register block 1 (DMR1) [p. 11-7]			

NOTE:

External masters cannot access MCF5307 on-chip memories or MBAR, but they can access DRAM controller registers.

For More Information On This Product,
Go to: www.freescale.com
Freescale Semiconductor, Inc.

Table 11-5. DMR0/DMR1 Field Descriptions (Continued)

Bits	Name	Description
6–1	AMx	Address modifier masks. Determine which accesses can occur in a given DRAM block. 0 Allow access type to hit in DRAM 1 Do not allow access type to hit in DRAM
0	V	Valid. Cleared at reset to ensure that the DRAM block is not erroneously decoded. 0 Do not decode DRAM accesses. 1 Registers controlling the DRAM block are initialized; DRAM accesses can be decoded.

Bit	Associated Access Type	Access Definition
C/I	CPU space/interrupt acknowledge	MOVEC instruction or interrupt acknowledge cycle
AM	Alternate master	External or DMA master
SC	Supervisor code	Any supervisor-only instruction access
SD	Supervisor data	Any data fetched during the instruction access
UC	User code	Any user instruction
UD	User data	Any user data

11.3.3 General Asynchronous Operation Guidelines

The DRAM controller provides control for $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{DRAMW}}$ signals, as well as address multiplexing and bus cycle termination. Whether the mode is synchronous or asynchronous determines signal control and termination. To reduce complexity, multiplexing is the same for both modes. Table 11-6 shows the scheme for DRAM configurations. This scheme works for symmetric configurations (in which the number of rows equals the number of columns) as well as asymmetric configurations (in which the number of rows and columns are different).

Table 11-6. Generic Address Multiplexing Scheme

Address Pin	Row Address	Column Address	Notes Relating to Port Sizes
17	17	0	8-bit port only
16	16	1	8- and 16-bit ports only
15	15	2	
14	14	3	
13	13	4	
12	12	5	
11	11	6	
10	10	7	
9	9	8	
17	17	16	32-bit port only
18	18	17	16-bit port only or 32-bit port with only 8 column address lines
19	19	18	16-bit port only when at least 9 column address lines are used

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Part III
Peripheral Module

Intended Audience

Part III describes the operation and configuration of the MCF5307 DMA, timer, UART, and parallel port modules, and describes how they interface with the system integration unit, described in Part II.

Contents

Part III contains the following chapters:

- Chapter 12, “DMA Controller Module,” provides an overview of the DMA controller module and describes in detail its signals and registers. The latter sections of this chapter describe operations, features, and supported data transfer modes in detail, showing timing diagrams for various operations.
- Chapter 13, “Timer Module,” describes configuration and operation of the two general-purpose timer modules, timer 0 and timer 1. It includes programming examples.
- Chapter 14, “UART Modules,” describes the use of the universal asynchronous/synchronous receiver/transmitters (UARTs) implemented on the MCF5307 and includes programming examples.
- Chapter 15, “Parallel Port (General-Purpose I/O),” describes the operation and programming model of the parallel port pin assignment, direction-control, and data registers. It includes a code example for setting up the parallel port.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

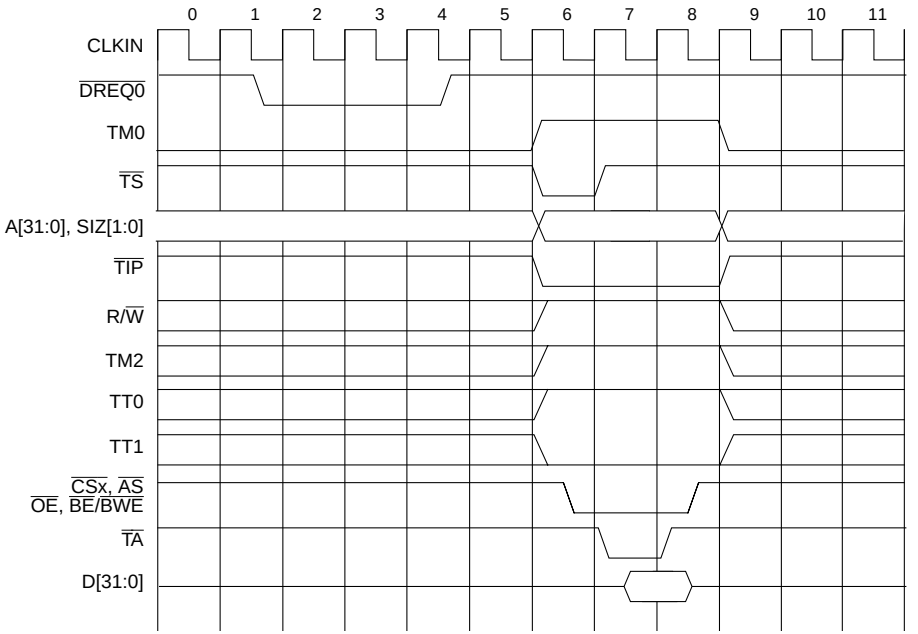

Figure 12-13. Single-Address DMA Transfer

12.5.4.2 Auto-Alignment

Auto-alignment allows block transfers to occur at the optimal size based on the address, byte count, and programmed size. To use this feature, DCR[AA] must be set. The source is auto-aligned if SSIZE indicates a transfer size larger than DSIZE. Source alignment takes precedence over the destination when the source and destination sizes are equal. Otherwise, the destination is auto-aligned. The address register chosen for alignment increments regardless of the increment value. Configuration error checking is performed on registers not chosen for alignment.

If BCR is greater than 16, the address determines transfer size. Bytes, words, or longwords are transferred until the address is aligned to the programmed size boundary, at which time accesses begin using the programmed size.

If BCR is less than 16 at the start of a transfer, the number of bytes remaining dictates transfer size. For example, AA = 1, SAR = 0x0001, BCR = 0x00F0, SSIZE = 00 (longword), and DSIZE = 01 (byte). Because SSIZE > DSIZE, the source is auto-aligned. Error checking is performed on destination registers. The access sequence is as follows:

1. Read byte from 0x0001—write 1 byte, increment SAR.
2. Read word from 0x0002—write 2 bytes, increment SAR.
3. Read longword from 0x0004—write 4 bytes, increment SAR.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Table 14-3. UMR2*n* Field Descriptions (Continued)

Bits	Name	Description
3–0	SB	Stop-bit length control. Selects the length of the stop bit appended to the transmitted character. Stop-bit lengths of 9/16th to 2 bits are programmable for 6–8 bit characters. Lengths of 1 1/16th to 2 bits are programmable for 5-bit characters. In all cases, the receiver checks only for a high condition at the center of the first stop-bit position, that is, one bit time after the last data bit or after the parity bit, if parity is enabled. If an external 1x clock is used for the transmitter, clearing bit 3 selects one stop bit and setting bit 3 selects 2 stop bits for transmission.

SB	5 Bits	6–8 Bits	SB	5 Bits	6–8 Bits	SB	5–8 Bits	SB	5–8 Bits
0000	1.063	0.563	0100	1.313	0.813	1000	1.563	1100	1.813
0001	1.125	0.625	0101	1.375	0.875	1001	1.625	1101	1.875
0010	1.188	0.688	0110	1.438	0.938	1010	1.688	1110	1.938
0011	1.250	0.750	0111	1.500	1.000	1011	1.750	1111	2.000

14.3.3 UART Status Registers (USR*n*)

The USR*n*, Figure 14-4, shows status of the transmitter, the receiver, and the FIFO.

	7	6	5	4	3	2	1	0
Field	RB	FE	PE	OE	TxEMP	TxRDY	FFULL	RxRDY
Reset	0000_0000							
R/W	Read only							
Address	MBAR + 0x1C4 (USR0), 0x204 (USR1)							

Figure 14-4. UART Status Register (USR*n*)

Table 14-4 describes USR*n* fields.

Table 14-4. USR*n* Field Descriptions

Bits	Name	Description
7	RB	Received break. The received break circuit detects breaks that originate in the middle of a received character. However, a break in the middle of a character must persist until the end of the next detected character time. 0 No break was received. 1 An all-zero character of the programmed length was received without a stop bit. RB is valid only when RxRDY = 1. Only a single FIFO position is occupied when a break is received. Further entries to the FIFO are inhibited until RxD returns to the high state for at least one-half bit time, which is equal to two successive edges of the UART clock.
6	FE	Framing error. 0 No framing error occurred. 1 No stop bit was detected when the corresponding data character in the FIFO was received. The stop-bit check occurs in the middle of the first stop-bit position. FE is valid only when RxRDY = 1.

Freescale Semiconductor, Inc.

The receiver detects the beginning of a break in the middle of a character if the break persists through the next character time. If the break begins in the middle of a character, the receiver places the damaged character in the Rx FIFO stack and sets the corresponding USR*n* error bits and USR*n*[RxRDY]. Then, if the break lasts until the next character time, the receiver places an all-zero character into the Rx FIFO and sets USR*n*[RB,RxRDY].

14.5.2.3 FIFO Stack

The FIFO stack is used in the UART's receiver buffer logic. The stack consists of three receiver holding registers. The receive buffer consists of the FIFO and a receiver shift register connected to the RxD (see Figure 14-20). Data is assembled in the receiver shift register and loaded into the top empty receiver holding register position of the FIFO. Thus, data flowing from the receiver to the CPU is quadruple-buffered.

In addition to the data byte, three status bits, parity error (PE), framing error (FE), and received break (RB), are appended to each data character in the FIFO; OE (overrun error) is not appended. By programming the ERR bit in the channel's mode register (UMR1*n*), status is provided in character or block modes.

USR*n*[RxRDY] is set when at least one character is available to be read by the CPU. A read of the receiver buffer produces an output of data from the top of the FIFO stack. After the read cycle, the data at the top of the FIFO stack and its associated status bits are popped and the receiver shift register can add new data at the bottom of the stack. The FIFO-full status bit (FFULL) is set if all three stack positions are filled with data. Either the RxRDY or FFULL bit can be selected to cause an interrupt.

The two error modes are selected by UMR1*n*[ERR] as follows:

- In character mode (UMR1*n*[ERR] = 0, status is given in the USR*n* for the character at the top of the FIFO.
- In block mode, the USR*n* shows a logical OR of all characters reaching the top of the FIFO stack since the last RESET ERROR STATUS command. Status is updated as characters reach the top of the FIFO stack. Block mode offers a data-reception speed advantage where the software overhead of error-checking each character cannot be tolerated. However, errors are not detected until the check is performed at the end of an entire message—the faulting character is not identified.

In either mode, reading the USR*n* does not affect the FIFO. The FIFO is popped only when the receive buffer is read. The USR*n* should be read before reading the receive buffer. If all three receiver holding registers are full, a new character is held in the receiver shift register until space is available. However, if a second new character is received, the contents of the the character in the receiver shift register is lost, the FIFOs are unaffected, and USR*n*[OE] is set when the receiver detects the start bit of the new overrunning character.

To support flow control, the receiver can be programmed to automatically negate and assert $\overline{\text{RTS}}$, in which case the receiver automatically negates $\overline{\text{RTS}}$ when a valid start bit is detected and the FIFO stack is full. The receiver asserts $\overline{\text{RTS}}$ when a FIFO position becomes

For More Information On This Product,
Go to: www.freescale.com
Freescale Semiconductor, Inc.

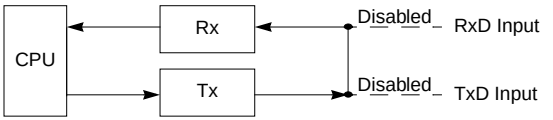

Figure 14-24. Local Loop-Back

Features of this local loop-back mode are as follows:

- Transmitter and CPU-to-receiver communications continue normally in this mode.
- RxD input data is ignored
- TxD is held marking
- The receiver is clocked by the transmitter clock. The transmitter must be enabled, but the receiver need not be.

14.5.3.3 Remote Loop-Back Mode

In remote loop-back mode, shown in Figure 14-25, the channel automatically transmits received data bit by bit on the TxD output. The local CPU-to-transmitter link is disabled. This mode is useful in testing receiver and transmitter operation of a remote channel. For this mode, the transmitter uses the receiver clock.

Because the receiver is not active, received data cannot be read by the CPU and error status conditions are inactive. Received parity is not checked and is not recalculated for transmission. Stop bits are sent as they are received. A received break is echoed as received until the next valid start bit is detected.

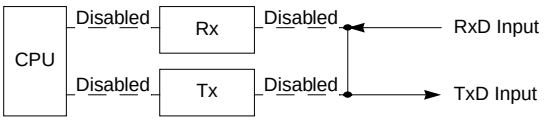

Figure 14-25. Remote Loop-Back

14.5.4 Multidrop Mode

Setting UMR1*n*[PM] programs the UART to operate in a wake-up mode for multidrop or multiprocessor applications. In this mode, a master can transmit an address character followed by a block of data characters targeted for one of up to 256 slave stations.

Although slave stations have their channel receivers disabled, they continuously monitor the master's data stream. When the master sends an address character, the slave receiver channel notifies its respective CPU by setting USR*n*[RxRDY] and generating an interrupt (if programmed to do so). Each slave station CPU then compares the received address to its station address and enables its receiver if it wishes to receive the subsequent data characters or block of data from the master station. Slave stations not addressed continue monitoring the data stream. Data fields in the data stream are separated by an address character. After

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 16-3. Pins 105–156 (Right, Bottom-to-Top) (Continued)

Pin		Alternate Function	I/O	Description	Drive (mA)
No	Name				
142	D4	ADDR_CONF	I/O	Data bus/Address configuration	8
143	D3	FREQ1	I/O	Data bus/CLKIN Frequency	8
144	D2	FREQ0	I/O	Data bus/CLKIN Frequency	8
145	VCC	—	—	Power input	—
146	D1	DIVIDE1	I/O	Data bus/Divide control PCLK:BCLKO	8
147	D0	DIVIDE0	I/O	Data bus/Divide control PCLK:BCLKO	8
148	GND	—	—	Ground pin	—
149	DSCLK	$\overline{\text{TRST}}$	I	Debug serial clock/JTAG Reset	—
150	TCK	TCK	I	JTAG clock	—
151	DSO	TDO	O	Debug serial out/JTAG data out	8
152	VCC	—	—	Power input	—
153	DSI	TDI	I	Debug serial input/JTAG data in	—
154	$\overline{\text{BKPT}}$	TMS	I	Debug breakpoint/JTAG mode select	—
155	$\overline{\text{HIZ}}$	—	I	High impedance override	—
156	GND	—	—	Ground pin	—

Table 16-4. Pins 157–208 (Top, Right-to-Left)

Pin		Alternate Function	I/O	Description	Drive (mA)
No	Name				
157	VCC	—	—	Power input	—
158	$\overline{\text{CTS1}}$	—	I	UART1 clear-to-send	—
159	$\overline{\text{RTS1}}$	—	O	UART1 request-to-send	8
160	RXD1	—	I	UART1 receive data	—
161	TXD1	—	O	UART1 transmit data	8
162	GND	—	—	Ground pin	—
163	$\overline{\text{CTS0}}$	—	I	UART0 clear-to-send	—
164	$\overline{\text{RTS0}}$	—	O	UART0 request-to-send	8
165	RXD0	—	I	UART0 receive data	—
166	TXD0	—	O	UART0 transmit data	8
167	VCC	—	—	Power input	—
168	EDGESEL	—	I	SDRAM bus clock edge select	—
169	GND	—	—	Ground pin	—
170	BCLKO	—	O	Bus clock output	16

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 17-2. MCF507 Alphabetical Signal Index (Continued)

Abbreviation	Signal Name	Function	I/O	Page
$\overline{\text{SRAS}}$	Synchronous row address strobe	DRAM	O	17-17
$\overline{\text{TA}}$	Transfer acknowledge	Bus	I/O	17-9
TCK	Test clock	JTAG	I	17-23
TDI/DSI	Test data input/Development serial input	JTAG	I	17-22
TDO/DSO	Test data output/Development serial output	JTAG	O	17-22
TIN[1:0]	Timer input	Timer	I	17-19
$\overline{\text{TIP}}$	Transfer in progress	Bus	O	17-10
TMS/$\overline{\text{BKPT}}$	Test mode select/Breakpoint	JTAG	I	17-22
TM[2:0]	Transfer modifier	Bus	O	17-10
TOUT[1:0]	Timer outputs	Timer	O	17-19
$\overline{\text{TRST}}$/DSCLK	Test reset/Development serial clock	JTAG	I	17-21
$\overline{\text{TS}}$	Transfer start	Bus	I/O	17-9
TT[1:0]	Transfer type	Bus	O	17-10
TxD[1:0]	Transmit data	Serial module	O	17-18

17.2 MCF5307 Bus Signals

The bus signals provide the external bus interface to the MCF5307.

17.2.1 Address Bus

The address bus provides the address of the byte or most-significant byte (MSB) of the word or longword being transferred. The address lines also serve as the DRAM addressing, providing multiplexed row and column address signals. When an external device has ownership of the MCF5307 bus, the device must drive the address bus and assert $\overline{\text{TS}}$ or $\overline{\text{AS}}$ to indicate the start of a bus cycle. During an interrupt acknowledge access, A[4:2] indicate the interrupt level being acknowledged.

17.2.1.1 Address Bus (A[23:0])

The lower 24 bits of the address bus become valid when $\overline{\text{TS}}$ is asserted. A[4:2] indicate the interrupt level during interrupt acknowledge cycles.

17.2.1.2 Address Bus (A[31:24]/PP[15:8])

These multiplexed pins can serve as the most-significant byte of the address bus, or as the most-significant byte of the parallel port. Programming the PAR in the system integration module (SIM) determines the function of each of these eight multiplexed pins. These pins are programmable on a bit-by-bit basis.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

controller in test logic reset state immediately. Tying it to V_{DD} causes the JTAG controller (if TMS is a logic level of 1) to eventually enter test logic reset state after 5 TCK clocks.

If MTMOD0 is low, DSCLK is selected. DSCLK is the development serial clock for the serial interface to the debug module. The maximum DSCLK frequency is 1/5 CLKIN. See Chapter 5, “Debug Support.”

17.14.2 Test Mode Select/Breakpoint (TMS/$\overline{\text{BKPT}}$)

If MTMOD0 is high, TMS is selected. The TMS input provides information to determine the JTAG test operation mode. The state of TMS and the internal 16-state JTAG controller state machine at the rising edge of TCK determine whether the JTAG controller holds its current state or advances to the next state. This directly controls whether JTAG data or instruction operations occur. TMS has an internal pull-up resistor so that if it is not driven low, it defaults to a logic level of 1. But if TMS is not used, it should be tied to V_{DD}.

If MTMOD0 is low, $\overline{\text{BKPT}}$ is selected. $\overline{\text{BKPT}}$ signals a hardware breakpoint to the processor in debug mode. See Chapter 5, “Debug Support.”

17.14.3 Test Data Input/Development Serial Input (TDI/DSI)

If MTMOD0 is high, TDI is selected. TDI provides the serial data port for loading the various JTAG boundary scan, bypass, and instruction registers. Shifting in data depends on the state of the JTAG controller state machine and the instruction in the instruction register. Shifts occur on the TCK rising edge. TDI has an internal pull-up resistor, so when not driven low it defaults to high. But if TDI is not used, it should be tied to V_{DD}.

If MTMOD0 is low, DSI is selected. DSI provides the single-bit communication for debug module commands. See Chapter 5, “Debug Support.”

17.14.4 Test Data Output/Development Serial Output (TDO/DSO)

If MTMOD0 is high, TDO is selected. The TDO output provides the serial data port for outputting data from JTAG logic. Shifting out data depends on the JTAG controller state machine and the instruction in the instruction register. Data shifting occurs on the falling edge of TCK. When TDO is not outputting test data, it is three-stated. TDO can be three-stated to allow bused or parallel connections to other devices having JTAG.

If MTMOD0 is low, DSO is selected. DSO provides single-bit communication for debug module responses. See Chapter 5, “Debug Support.”

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

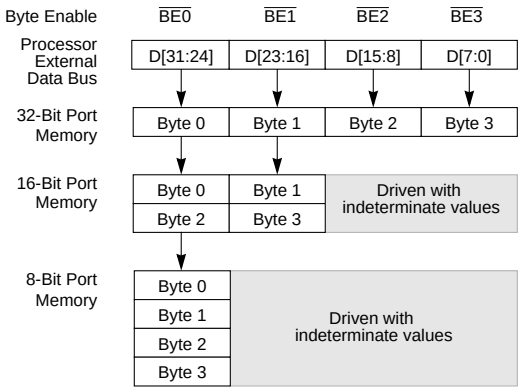

Figure 18-2. Connections for External Memory Port Sizes

The timing relationships between BCLKOchip select ($\overline{CS}$[7:0]), byte enable/byte write enables ($\overline{BE}$/$\overline{BWE}$[3:0]), and output enable ($\overline{OE}$) are similar to their relationships with address strobe ($\overline{AS}$) in that all transitions occur during the low phase of BCLKO. However, as shown in Figure 18-3, differences in on-chip signal routing and external loading may prevent signals from asserting simultaneously.

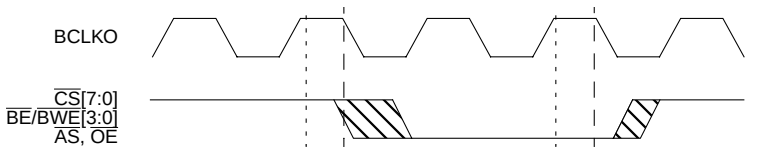

Figure 18-3. Chip-Select Module Output Timing Diagram

18.4.1 Bus Cycle Execution

When a bus cycle is initiated, the MCF5307 first compares its address with the base address and mask configurations programmed for chip selects 0–7 (CSCR0–CSCR7) and for DRAM blocks 0 and 1 address and control registers (DACR0 and DACR1). If the driven address matches a programmed chip select or DRAM block, the appropriate chip select is asserted or the DRAM block is selected using the specifications programmed in the respective configuration register. Otherwise, the following occurs:

- If the address and attributes do not match in CSCR or DACR, the MCF5307 runs an external burst-inhibited bus cycle with a default of external termination on a 32-bit port.
- If an address and attribute match in multiple CSCRs, the matching chip-select signals are driven; however, the MCF5307 runs an external burst-inhibited bus cycle with external termination on a 32-bit port.
- If an address and attribute match both DACRs or a DACR and a CSCR, the operation is undefined.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

transfer.

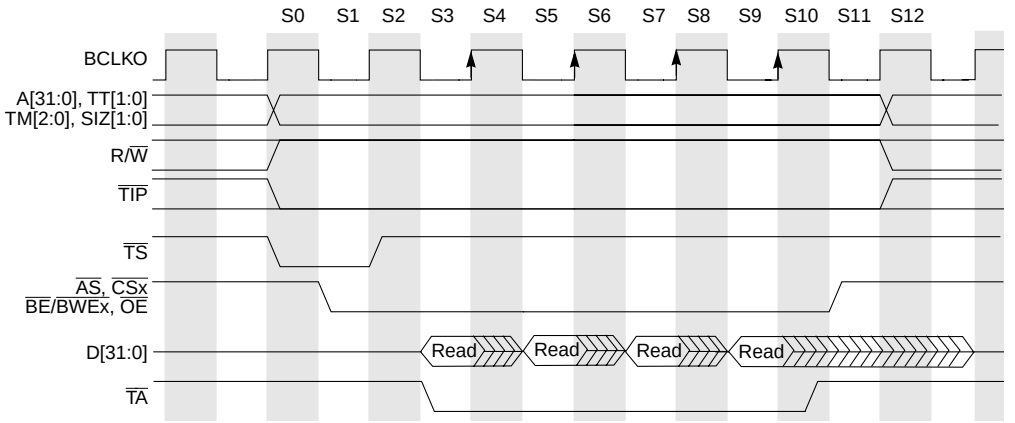

Figure 18-19. Longword Read from an 8-Bit Port, External Termination

Note that with external termination, address signals do not change. With internal termination, Figure 18-20, A[1:0] increment for the same longword transfer.

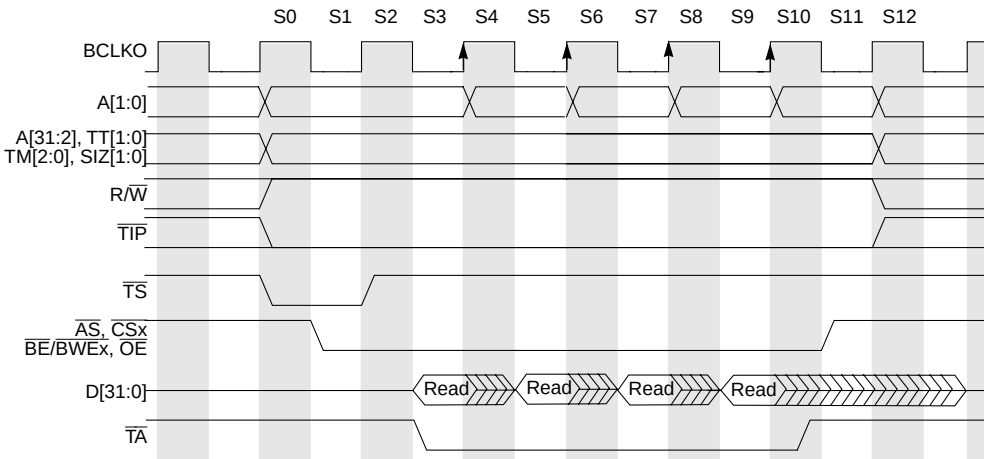

Figure 18-20. Longword Read from an 8-Bit Port, Internal Termination

18.5 Misaligned Operands

Because operands, unlike opcodes, can reside at any byte boundary, they are allowed to be misaligned. A byte operand is properly aligned at any address, a word operand is misaligned at an odd address, and a longword is misaligned at an address not a multiple of four. Although the MCF5307 enforces no alignment restrictions for data operands (including program counter (PC) relative data addressing), additional bus cycles are required for misaligned operands.

For More Information On This Product,
Go to: www.freescale.com
Freescale Semiconductor, Inc.

INDEX

S

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.