SECTION 5
SRAM

5.1 SRAM FEATURES

- 512-Byte SRAM, Organized as 128 x 32 Bits
- Single-Cycle Access
- Physically Located on ColdFire core's High-Speed Local Bus
- Byte, Word, Longword Address Capabilities
- Memory Mapping Defined by the Customer

5.2 SRAM OPERATION

The SRAM module provides a general-purpose memory block that the ColdFire core can access in a single cycle. You can specify the location of the memory block to any 0-modulo-512 address within the four gigabyte address space. The memory is ideal for storing critical code or data structures, or for use as the system stack. Because the SRAM module is physically connected to the processor's high-speed local bus, it can service core-initiated accesses, or memory-referencing commands from the debug module.

Depending on configuration information, instruction fetches can be sent to both the instruction cache and the SRAM block simultaneously. If the instruction fetch address is mapped into the region defined by the SRAM, the SRAM provides the data back to the processor, and the I-Cache data is discarded. Accesses from the SRAM module are not cached.

5.3 PROGRAMMING MODEL

5.3.1 SRAM Register Memory Map

Table 5-1 below shows the memory map of the SRAM register.

The following lists several keynotes regarding the programming model table:

- The SRAM base address register can only be accessed in supervisor mode using the MOVEC instruction with an Rc value of $C04.
- Addresses not assigned to the register and undefined register bits are reserved for future expansion. Write accesses to these reserved address spaces and reserved register bits have no effect; read accesses will return zeros.
- The reset value column indicates the register initial value at reset. Certain registers can be uninitialized at reset, i.e., they may contain random values after reset.

- The access column indicates if the corresponding register allows both read/write functionality (R/W), read-only functionality (R), or write-only functionality (W). If a read access to a write-only register is attempted, zeros will be returned. If a write access to a read-only register is attempted the access will be ignored and no write will occur.

Table 5-1. Memory Map of SIM Registers

ADDRESS	NAME	WIDTH	DESCRIPTION	RESET VALUE	ACCESS
MOVEC with $C04	RAMBAR	32	SRAM Base Address Register	uninitialized (except V=0)	W

5.3.2 SRAM Registers

5.3.2.1 SRAM BASE ADDRESS REGISTER (RAMBAR). The RAMBAR determines the base address location of the internal SRAM module, as well as the definition of the types of accesses that are allowed for it.

The RAMBAR is a 32-bit write-only supervisor control register. It is accessed in the CPU address space via the MOVEC instruction with an Rc encoding of $C04. The RAMBAR can be read when in Background Debug mode (BDM). At system reset, the V bit is cleared and the remainder bits of the RAMBAR are uninitialized. To access the SRAM module, RAMBAR should be written with the appropriate base address after system reset.

SRAM Base Address Register(RAMBAR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
RESET: -	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA15	BA14	BA13	BA12	BA11	BA10	BA9	WP	-	-	C/I	SC	SD	UC	UD	V
RESET: -	-	-	-	-	-	-	-	0	0	-	-	-	-	-	0

BA31-BA9 - Base Address

This field defines the base address location of the SRAM module. By programming this field, you can locate the SRAM anywhere within the four gigabyte ColdFire core address space.

WP - Write Protect

This field allows only read accesses to the SRAM. When this bit is set, any attempted write access will generate an access error exception in the MCF5206.

0 = Allow read and write accesses to the SRAM module
1 = Allow only read accesses to the SRAM module

C/I, SC, SD, UC, UD - Address Space Masks

This field allows specific address spaces to be enabled or disabled, placing the internal modules in a specific address space. If an address space is disabled, an access to the register location in that address space becomes an external bus access, and the module resource is not accessed. The address space mask bits are:

C/I = CPU Space/Interrupt Acknowledge Cycle mask
SC = Supervisor Code address space mask
SD = Supervisor Data address space mask
UC = User Code address space mask
UD = User Data address space mask

For each address space bit:

0= An access to the internal module can occur for this address space.
1= Disable this address space from the internal module selection. If this address space is used, no access will occur to the internal peripheral, instead an external bus cycle will be generated.

V - Valid

This bit indicates when the contents of the RAMBAR are valid. The base address value is not used, and the SRAM module is not accessible until the V bit is set. An external bus cycle is generated if the base address field matches the internal core address, and the V bit is clear.

0 = Contents of RAMBAR are not valid.
1 = Contents of RAMBAR are valid.

The mapping of a given access into the SRAM uses the following algorithm to determine if the access “hits” in the memory:

```
if (RAMBAR[0] = 1)
    if (requested address[31:9] = RAMBAR[31:9])
        if (address space mask of the requested type = 0)
            Access is mapped to the SRAM module
            if (access = read)
                Read the SRAM and return the data
            if (access = write)
                if (RAMBAR[8] = 0)
                     Write the data into the SRAM
                else Signal a write-protect access error
```

5.3.3 SRAM Initialization

After a hardware reset, the contents of the SRAM module are undefined. The valid bit of the RAMBAR is cleared, disabling the module. If the SRAM needs to be initialized with instructions or data, you should perform the following steps:

1. Load the RAMBAR mapping the SRAM module to the desired location within the

address space.

2. Read the source data and write it to the SRAM. There are various instructions to support this function, including memory-to-memory move instructions, or the MOVEM opcode. The MOVEM instruction is optimized to generate line-sized burst fetches on 0-modulo-16 addresses, so this opcode generally provides maximum performance.
3. After the data has been loaded into the SRAM, it may be appropriate to load a revised value into the RAMBAR with a new set of “attributes.” These attributes consist of the write-protect and address space mask fields.

The ColdFire processor or an external emulator using the Debug module can perform these initialization functions.

5.3.4 Power Management

As noted previously, depending upon the configuration defined by the RAMBAR, instruction fetch accesses can be sent to the SRAM module and the I-Cache simultaneously. If the access is mapped to the SRAM module, it sources the read data, discarding the I-Cache access. If the SRAM is used only for data operands, setting the SC and UC mask bits in the RAMBAR to 1 will lower power dissipation. This will disable the SRAM during all instruction fetches. Additionally, if the SRAM contains only instructions, setting the SD and UD mask bits in the RAMBAR to 1 masking operand accesses will reduce power dissipation.

Consider the examples on Table 5-2 of typical RAMBAR settings:

Table 5-2. Examples of Typical RAMBAR Settings

DATA CONTAINED IN SRAM	RAMBAR[7:0]
Code only	$2B
Data only	$35
Both Code and Data	$21