SECTION 4
INSTRUCTION CACHE

4.1 FEATURES OF INSTRUCTION CACHE

- 512-Byte Direct-Mapped Cache
- Single-Cycle Access on Cache Hits
- Physically Located on ColdFire core's High-Speed Local Bus
- Nonblocking Design to Maximize Performance
- 16-Byte Line-Fill Buffer
- Configurable Cache-Miss Fetch Algorithm

4.2 INSTRUCTION CACHE PHYSICAL ORGANIZATION

The instruction cache is a direct-mapped single-cycle memory, organized as 32 lines, each containing 16 bytes. The memory storage consists of a 32-entry tag array (containing addresses and a valid bit), and the data array containing 512 bytes of instruction data, organized as 128 x 32 bits.

The two memory arrays are accessed in parallel: bits [8:4] of the instruction fetch address provide the index into the tag array, and bits [8:2] addressing the data array. The tag array outputs the address mapped to the given cache location along with the valid bit for the line. This address field is compared to bits [31:9] of the instruction fetch address from the local bus to determine if a cache hit in the memory array has occurred. If the desired address is mapped into the cache memory, the output of the data array is driven onto the ColdFire core's local data bus completing the access in a single cycle.

The tag array maintains a single valid bit per line entry. Accordingly, only entire 16-byte lines are loaded into the instruction cache.

The instruction cache also contains a 16-byte fill buffer that provides temporary storage for the last line fetched in response to a cache miss. With each instruction fetch, the contents of the line fill buffer are examined. Thus, each instruction fetch address examines both the tag memory array and the line fill buffer to see if the desired address is mapped into either hardware resource, with the linefill buffer having priority over the instruction cache. A cache hit in either the memory array or the line-fill buffer is serviced in a single cycle. Because the line fill buffer maintains valid bits on a longword basis, hits in the buffer can be serviced immediately without waiting for the entire line to be fetched.

If the referenced address is not contained in the memory array or the line-fill buffer, the instruction cache initiates the required external fetch operation. In most situations, this is a 16-byte line-sized burst reference.

The hardware implementation is a nonblocking design, meaning the ColdFire core's local bus is released after the initial access of a miss. Thus, the cache or the SRAM module can service subsequent requests while the remainder of the line is being fetched and loaded into the fill buffer.

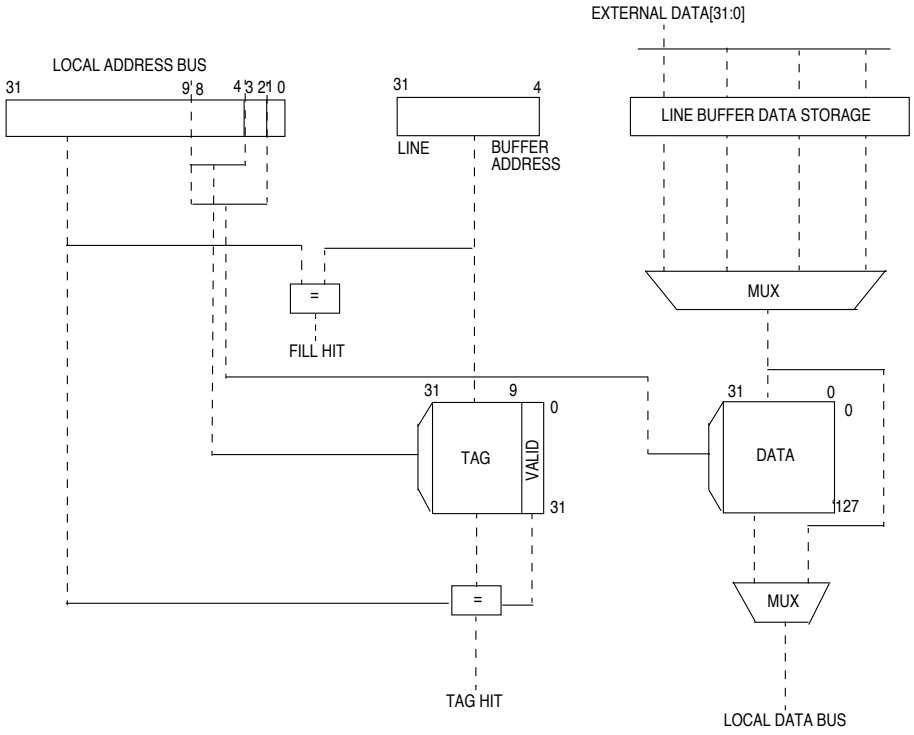

Figure 4-1. Instruction Cache Block Diagram

4.3 INSTRUCTION CACHE OPERATION

The instruction cache is physically connected to the ColdFire core's local bus, allowing it to service all instruction fetches from the ColdFire core and certain memory fetches initiated by the debug module. Typically, the Debug module's memory references appear as supervisor data accesses, but the unit can be programmed to generate user-mode accesses and/or instruction fetches. The instruction cache processes any instruction fetch access in the normal manner.

4.3.1 Interaction With Other Modules

Since the instruction cache and high-speed SRAM module are connected to the ColdFire core's local data bus, certain user-defined configurations can result in simultaneous instruction fetch processing.

If the referenced address is mapped into the SRAM module, that module will service the request in a single cycle. In this case, data accessed from the instruction cache is simply discarded, and no external memory references are generated. If the address is not mapped into the SRAM space, the instruction cache handles the request in the normal fashion.

4.3.2 Memory Reference Attributes

For every memory reference the ColdFire core or the Debug module generates, a set of “effective attributes” is determined based on the address and the Access Control Registers (ACR0, ACR1). This set of attributes includes the cacheable/noncacheable definition, the precise/imprecise handling of operand write, and the write-protect capability.

In particular, each address is compared to the values programmed in the Access Control Registers (ACR). If the address matches one of the ACR values, the access attributes from that ACR are applied to the reference. If the address does not match either ACR, then the default value defined in the Cache Control Register (CACR) is used. The specific algorithm is as follows:

```
if (address = ACR0_address including mask)
     Effective Attributes = ACR0 attributes
else if (address = ACR1_address including mask)
          Effective Attributes = ACR1 attributes
     else Effective Attributes = CACR default attributes
```

4.3.3 Cache Coherency and Invalidation

The instruction cache does not monitor ColdFire core data references for accesses to cached instructions, therefore software must maintain cache coherency by invalidating the appropriate cache entries after modifying code segments.

The cache invalidation can be performed in two ways. The assertion of bit 24 in the CACR, via a CPU space write, forces the entire instruction cache to be marked as invalid. The invalidation operation requires 32 cycles because the cache sequences through the entire tag array, clearing a single location each cycle. Any subsequent instruction fetch accesses are postponed until the invalidation sequence is complete.

The privileged CPUSHL instruction can invalidate a single cache line. When this instruction is executed, the cache entry defined by bits[8:4] of the source address register is invalidated, provided bit 28 of the CACR is cleared.

These invalidation operations may be initiated from the ColdFire core or the debug module.

4.3.4 RESET

A hardware reset clears the CACR disabling the instruction cache. The contents of the tag array are not affected by the reset. Accordingly, the system startup code must explicitly perform a cache invalidation by setting CACR[24] before the cache can be enabled.

4.3.5 Cache Miss Fetch Algorithm/Line Fills

As discussed in **Section 4.2 Instruction cache Physical Organization**, the instruction cache hardware includes a 16-byte line fill buffer for providing temporary storage for the last fetched instruction.

With the cache enabled as defined by CACR[31], a cacheable instruction fetch that misses in both the tag memory and the line-fill buffer generates a external fetch. The size of the external fetch is determined by the value contained in the 2-bit CLNF field of the CACR and the miss address. Table 4-1 shows the relationship between the CLNF bits, the miss address, and the size of the external fetch.

Table 4-1. Initial Fetch Offset vs. CLNF Bits

CLNF[1:0]	LONGWORD ADDRESS BITS			
	00	01	10	11
00	Line	Line	Line	Longword
01	Line	Line	Longword	Longword
1X	Line	Line	Line	Line

Depending on the runtime characteristics of the application and the memory response speed, overall performance may be increased by programming the CLNF bits to values {00, 01}.

For all cases of a line-sized fetch, the critical longword defined by bits [3:2] of the miss address is accessed first, followed by the remaining three longwords that are accessed by incrementing the longword address in a modulo-16 fashion as shown below:

```
if miss address[3:2] = 00
   fetch sequence = {$0, $4, $8, $C}
if miss address[3:2] = 01
   fetch sequence = {$4, $8, $C, $0}
if miss address[3:2] = 10
   fetch sequence = {$8, $C, $0, $4}
if miss address[3:2] = 11
   fetch sequence = {$C, $0, $4, $8}
```

Once an external fetch has been initiated and the data loaded into the line-fill buffer, the instruction cache maintains a special “most-recently-used” indicator that tracks the

contents of the fill buffer versus its corresponding cache location. At the time of the miss, the hardware indicator is set, marking the fill buffer as “most recently used.” If a subsequent access occurs to the cache location defined by bits [8:4] of the fill buffer address, the data in the cache memory array is now most recently used, so the hardware indicator is cleared. In all cases, the indicator defines whether the contents of the line fill buffer or the memory data array are most recently used. At the time of the next cache miss, the contents of the line-fill buffer are written into the memory array if the entire line is present, and the fill buffer data is still most recently used compared to the memory array.

The fill buffer can also be used as temporary storage for line-sized bursts of non-cacheable references under control of CACR[10]. With this bit set, a noncacheable instruction fetch is processed as defined by Table 4-2. For this condition, the fill buffer is loaded and subsequent references can hit in the buffer, but the data is never loaded into the memory array.

Table 4-2 shows the relationship between CACR bits 31 and 10 and the type of instruction fetch.

Table 4-2. Instruction Cache Operation as Defined by CACR[31, 10]

CACR[31]	CACR[10]	TYPE OF INSTR. FETCH	DESCRIPTION
0	0	N/A	Instruction cache is completely disabled; all fetches are word, longword in size.
0	1	N/A	All fetches are word, longword in size
1	X	Cacheable	Fetch size is defined by Table 5-1 and contents of the line-fill buffer can be written into the memory array
1	0	Noncacheable	All fetches are longword in size, and not loaded into the line-fill buffer
1	1	Noncacheable	Fetch size is defined by Table 5-1 and loaded into the line-fill buffer, but are never written into the memory array.

4.4 INSTRUCTION CACHE PROGRAMMING MODEL

Three supervisor registers define the operation of the instruction cache and local bus controller: the Cache Control Register (CACR) and two Access Control Registers (ACR0, ACR1).

4.4.1 Instruction Cache Registers Memory Map

Table 4-3 below shows the memory map of the Instruction cache and access control registers.

The following lists several keynotes regarding the programming model table:

- The Cache Control Register and Access Control Registers can only be accessed in supervisor mode using the MOVEC instruction with an Rc value of $002, $004 and $005, respectively.

- Addresses not assigned to the registers 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 may be uninitialized upon 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 4-3. Memory Map of I-Cache Registers

ADDRESS	NAME	WIDTH	DESCRIPTION	RESET VALUE	ACCESS
MOVEC with $002	CACR	32	Cache Control Register	$0000	W
MOVEC with $004	ACR0	32	Access Control Register 0	$0000	W
MOVEC with $005	ACR1	32	Access Control Register 1	$0000	W

4.4.2 Instruction Cache Register

4.4.2.1 CACHE CONTROL REGISTER (CACR). The CACR controls the operation of the instruction cache. The CACR provides a set of default memory access attributes used when a reference address does not map into the spaces defined by the ACRs.

The CACR 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 $002. The CACR can be read when in Background Debug mode (BDM). At system reset, the entire register is cleared.

Cache Control Register (CACR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CENB	-	-	CPDI	CFRZ	-	-	CINV	-	-	-	-	-	-	-	-
RESET: 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	CEIB	DCM	DBWE	-	-	DWP	-	-	-	CLNF1	CLNF0
RESET: 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CENB - Cache Enable

When the cache is disabled, all instruction fetches generate word or longword-sized fetch. Generally, longword references are used for sequential fetches. If the processor branches to an odd word address, a word-sized fetch is generated. The memory array of the instruction cache is enabled only if CENB is asserted.

0 = Cache disabled
1 = Cache enabled

CPDI - Disable CPUSHL Invalidation

When the privileged CPUSHL instruction is executed, the cache entry defined by bits [8:4] of the address is invalidated if CPDI = 0. If CPDI = 1, no operation is performed.

0 = Enable invalidation
1 = Disable invalidation

CFRZ - Cache Freeze

This field allows the user to freeze the contents of the cache. When CFRZ is asserted, line fetches can be initiated and loaded into the line-fill buffer, but a valid cache entry is never overwritten. If a given cache location is invalid, the contents of the line-fill buffer can be written into the memory array while CFRZ is asserted.

0 = Normal Operation
1 = Freeze valid cache lines

CINV - Cache Invalidate

Setting this bit forces the cache to invalid each tag array entry. The invalidation process requires 32 machine cycles, with a single cache entry cleared per machine cycle. The state of this bit is always read as a zero. After a hardware reset, the cache must be invalidated before it is enabled.

0 = No operation
1 = Invalidate all cache locations

CIEB - Cache Enable Noncacheable Instruction Bursting

Setting this bit enables the line-fill buffer to be loaded with burst transfers under control of CLINF[1:0] for non-cacheable accesses. Noncacheable accesses are never written into the memory array.

0 = Disable burst fetches on noncacheable accesses
1 = Enable burst fetches on noncacheable accesses

DCM - Default Cache Mode

This bit defines the default cache mode: 0 is cacheable, 1 is noncacheable. For more information on the selection of the effective memory attributes, see **Section 4.3.2 Memory Reference Attributes**.

0 = Caching enabled
1 = Caching disabled

DBWE - Default Buffered Write Enable

This bit defines the default value for enabling buffered writes. If DBWE = 0, the termination of an operand write cycle on the processor's local bus is delayed until the external bus cycle is completed. If DBWE = 1, the write cycle on the local bus is terminated immediately and the operation buffered in the bus controller. In this mode, operand write cycles are effectively decoupled between the processor's local bus and the external bus.

Generally, enabled buffered writes provide higher system performance but recovery from access errors can be more difficult. For the ColdFire CPU, reporting access errors on operand writes is always imprecise and enabling buffered writes simply further decouples the write instruction from the signaling of the fault

0 = Disable buffered writes
1 = Enable buffered writes

DWP - Default Write Protection

0 = Read and write accesses permitted
1 = Only read accesses permitted

CLNF[1:0] - Cache Line Fill

These bits control the size of the memory request the cache issues to the bus controller for different initial line access offsets.

Table 4-4. External Fetch Size Based on Miss Address and CLNF

CLNF[1:0]	LONGWORD ADDRESS BITS/MISS ADDRESS			
	00	01	10	11
00	Line	Line	Line	Longword
01	Line	Line	Longword	Longword
10	Line	Line	Line	Line
11	Line	Line	Line	Line

4.4.2.2 ACCESS CONTROL REGISTERS (ACR0, ACR1). The ACRs control provide a definition of memory reference attributes for two memory regions (one per ACR). This set of effective attributes is defined for every memory reference using the ACRs or the set of default attributes contained in the CACR. The ACRs are examined for every memory reference that is NOT mapped to the SRAM module.

The ACRs are 32-bit write-only supervisor control register. They are accessed in the CPU address space via the MOVEC instruction with an Rc encoding of $004 and $005. The ACRs can be read when in background debug mode (BDM). At system reset, the entire registers are cleared.

Access Control Registers (ACR0, ACR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AB31	AB30	AB29	AB28	AB27	AB26	AB25	AB24	AM31	AM30	AM29	AM28	AM27	AM26	AM25	AM24
RESET: 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	SM1	SM0	-	-	-	-	-	-	CM	BUFW	-	-	WP	-	-
RESET: 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AB[31:24] - Address Base [31:24]

This 8-bit field is compared to address bits [31:24] from the processor's local bus under control of the ACR address mask. If the address matches, the attributes for the memory reference are sourced from the given ACR.

AM[31:24] - Address Mask [31:24]

This 8-bit field can mask any bit of the AB field comparison. If a bit in the AM field is set, then the corresponding bit of the address field comparison is ignored.

EN - Enable

The EN bit defines the ACR enable. Hardware reset clears this bit, disabling the ACR.

0 = ACR disabled
1 = ACR enabled

SM[1:0] - Supervisor mode

This two-bit field allows the given ACR to be applied to references based on operating privilege mode of the ColdFire processor. The field uses the ACR for user-references only, supervisor-references only, or all accesses.

00 = Match if user mode
01 = Match if supervisor mode
1x = Match always - ignore user/supervisor mode

CM - Cache Mode

This bit defines the cache mode: 0 is cacheable, 1 is noncacheable.

0 = Caching enabled
1 = Caching disabled

BWE- Buffered Write Enable

This bit defines the value for enabling buffered writes. If BWE = 0, the termination of an operand write cycle on the processor's local bus is delayed until the external bus cycle is completed. If BWE = 1, the write cycle on the local bus is terminated immediately, and the operation buffered in the bus controller. In this mode, operand write cycles are effectively decoupled between the processor's local bus and the external bus.

Generally, the enabling of buffered writes provides higher system performance, but recovery from access errors may be more difficult. For the ColdFire CPU, the reporting of access errors on operand writes is always imprecise, and enabling buffered writes simply decouples the write instruction from the signaling of the fault even more.

0 = Disable buffered writes
1 = Enable buffered writes

WP - Write Protect

The WP bit defines the write-protection attribute. If the effective memory attributes for a given access select the WP bit, any attempted write with this bit set is terminated with an access error.

0 = Read and write accesses permitted

1 = Only read accesses permitted