Hardware Architecture

The GSR is available in three configurations:

- **12008—**

  8-slot chassis with a 40-Gbps switch fabric.

- **12012—**

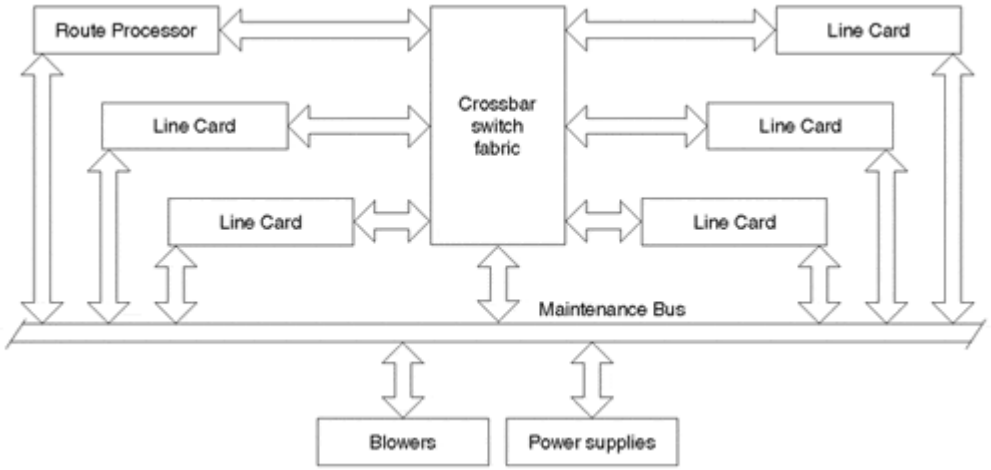
  12-slot chassis with a 60-Gbps switch fabric.

- **12016—**

  16-slot chassis with a 80-Gbps switch fabric that can be upgraded to 320 Gbps.

Figure 7-1 shows a high level overview of the Cisco 12000.

Figure 7-1. Cisco 1200 Architecture



The following lists the most important components of the Cisco 12000 architecture:

- Switching Fabric
- Maintenance Bus (MBUS)
- Route Processor
- Line Cards

Switching Fabric

All the Cisco router platforms discussed in this book so far are based on a shared bus architecture. The following items limit bus-based systems:

- **Arbitration Overhead—**

  Each time a Line Card (LC) wants to transmit onto a shared bus, it must enter into an arbitration phase that determines when it can transmit.

- **Single Card Transmission—**

  Only one LC can transmit onto the bus at any given time. No matter how many cards are in the system, only one card can transmit, and only one can receive, at any time.

The speed of a bus can be increased, within practical limits, by increasing the amount of data that can be transmitted at once (the *width* of the bus) and by increasing the speed at which the bus operates.

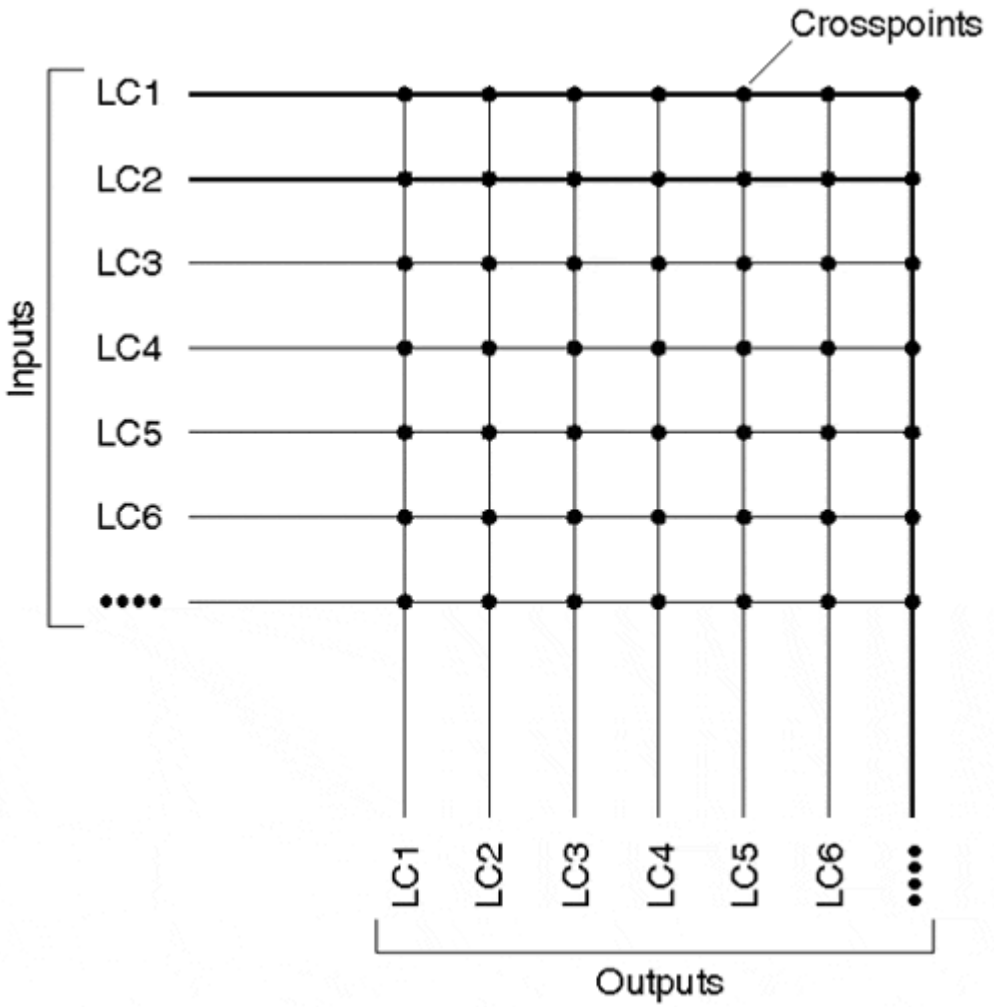
Shared memory architectures can be much faster. As you increase the speed of a shared memory system, you run into two scalability limits:

- Excessive overhead required to ensure no two processes write to the same shared memory location at the same time

- Physical limitations of memory speed

The Cisco 12000 eliminates both problems by using a crossbar switching fabric as its backplane. A crossbar is essentially 2N busses (where *N* is the number of LCs connected to the switching fabric), connected by N*N *crosspoints*. Figure 7-2 illustrates the basic concept. If one of the crosspoints is on, the two LCs connected through that crosspoint are connected and can communicate with one another.

Figure 7-2. Crossbar Switching Fabric



Although this crossbar switching fabric allows multiple LCs to transmit and receive data simultaneously, it's still possible (in fact, probable) that two cards might want to transmit data to a single outbound card at the same time. To resolve such conflicts, a scheduler is responsible for selecting which LCs transmit and which LCs receive during any given fabric cycle.

NOTE

The Cisco 12000 scheduler algorithm is based on a design from Stanford University. You can find a detailed description of switch fabric architectures in the paper, "Fast Switched Backplane for a Gigabit Switched Router," by Nick McKeown of Stanford University. You can find this paper on CCO at http://www.cisco.com/warp/public/cc/cisco/mkt/core/12000/tech/fasts_wp.pdf.

Physically, the Cisco 12000's switching fabric consists of up to five switch fabric cards. Two types of switch fabric cards exist in the system: the Clock Scheduler card (CSC) and the Switch Fabric Card (SFC). The CSC is a superset of the SFC.

The CSC and SFC both contain crossbar ASICs, which provide 1.25 Gb/s full duplex bandwidth per slot. The CSC also contains a scheduler ASIC, which grants Line Cards access to the fabric, and a clock, which provides a reference clock for all other cards in the system. Each system must have at least one working CSC; if two are installed, the second CSC provides clock and scheduler redundancy.

The output of the command, **show controller fia**, displays the active CSC, as demonstrated in Example 7-1

Example 7-1. *show controller fia* Displays the Active CSC

Router**#show controller fia** Fabric configuration: Full bandwidth nonredundant Master Scheduler: Slot 17 ….

Example 7-1 shows that the active CSC is in slot 17. The following section explains the meaning of fabric configuration (full bandwidth as shown in Example 7-1). This example also shows that there is only one CSC in the system because the fabric configuration is *nonredundant*.

Switch Fabric Bandwidth

The GSR switch fabric can run in two modes:

- Full bandwidth
- Quarter bandwidth

To figure out the bandwidth provided by the switching fabric in any given configuration, you have to keep in mind that each switching card provides 1.25-Gbps full-duplex bandwidth *per slot*. In full bandwidth mode, there are five fabric cards—two CSCs and three SFCs. Recall that the second CSC is in standby mode and provides clock and scheduler redundancy; therefore, only four of the switching cards are actually used to pass data across the fabric, so you would multiply 4 by 1.25 Gbps per switching card, and then multiply this result by the number of Line Cards installed (the maximum number depends on the platform).

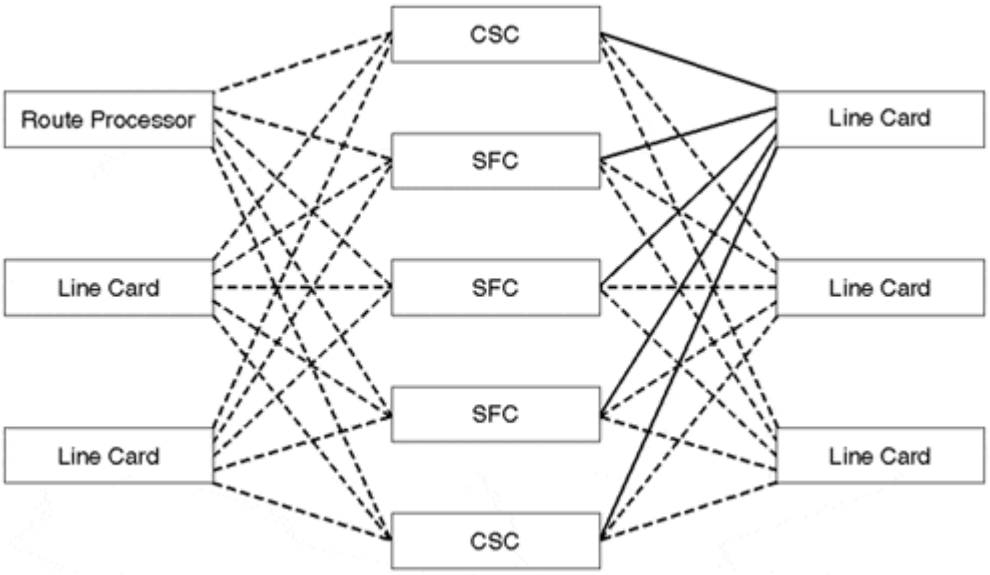
Table 7-1 documents the calculations to determine the total switch fabric bandwidth supported on each GSR configuration in full bandwidth mode.

Table 7-1. Determining Total Switch Fabric Bandwidth by GSR Configuration in Full Bandwidth Mode

GSR Configuration	Calculation	Total Switch Fabric Bandwidth Supported
12008	1.25 Gbps * 4 switching cards * 8 slots	40 Gbps
12012	1.25 Gbps * 4 switching cards * 12 slots	60 Gbps
12016	1.25 Gbps * 4 switching cards * 16 slots	80 Gbps

Note that in Figure 7-3, each Line Card connects to each switching card. In this configuration, traffic originating at a Line Card is segmented into equal size packets (64 bytes), divided into four 16-byte slices, and transmitted in parallel across the four connections to the switch cards. The fifth, redundant data path carries a CRC for the four slices to make possible error detection and some error recovery. If one of the four connections fails, the redundant data path is activated automatically with little or no loss of data. The unit of transfer across the switching fabric is always equal-size packets, also referred to as *Cisco cells*. This is covered in more detail in a later section in this chapter.

Figure 7-3. Cisco 12000 Switching Fabric



A *quarter bandwidth* configuration supports 1.25-Gbps full-duplex bandwidth per slot. Table 7-2 documents the calculations to determine the total switching fabric bandwidth for each configuration in quarter bandwidth mode.

Table 7-2. Determining Total Switch Fabric Bandwidth by GSR Configuration in Quarter Bandwidth Mode

GSR Configuration	Calculation	Total Switch Fabric Bandwidth Supported
12008	1.25 Gbps * 8 slots	10 Gbps
12012	1.25 Gbps * 12 slots	15 Gbps
12016	1.25 Gbps * 16 slots	20 Gbps

In the quarter bandwidth fabric configuration, traffic originating at a Line Card is segmented into equal-size cells, and then is sent on the only available connection to the active CSC.

NOTE

It's strongly recommended that you configure Cisco 12000s in full-bandwidth mode with redundancy. In fact, if the system has a high speed interface installed, such as a Gigabyte Ethernet, Fast Ethernet, DPT, OC-48/STM-16, or 4xOC12, full bandwidth mode is *mandatory*. Also note that the switching fabric offers bandwidth in only two modes—quarter or full—there is no half or three-quarter fabric bandwidth mode. Also, Cisco does not sell the quarter bandwidth mode.

The output of **show controller fia**(shown earlier in Example 7-1) displays the switch fabric mode in the system. In this example, the switch fabric is non-redundant, which means there is only one CSC in the system.

NOTE

The 12016 can be upgraded to an enhanced switch fabric based on a 64 x 64 crossbar that can provide up to 5-Gbps full-duplex bandwidth per switch fabric card per slot. The enhanced switch fabric in a full bandwidth mode provides 20 Gbps (5 Gbps * 4) bandwidth per slot, which is needed to support an OC-192 Line Card. The enhanced switch fabric is not available as of this writing.

Head of Line Blocking

Head of Line Blocking (HoLB) can severely reduce the throughput of a crossbar switch. Figure 7-4

illustrates HoLB.

Figure 7-4. Head of Line Blocking

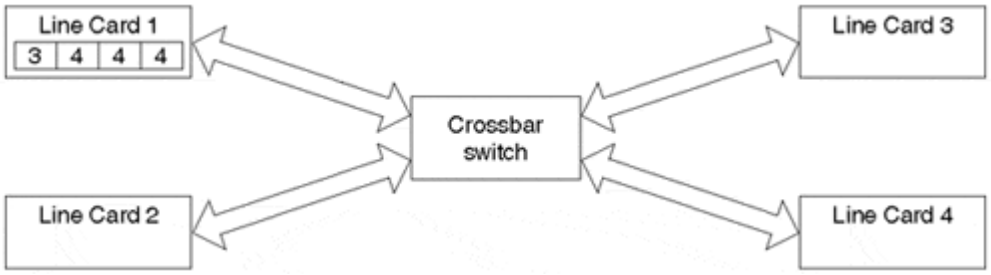
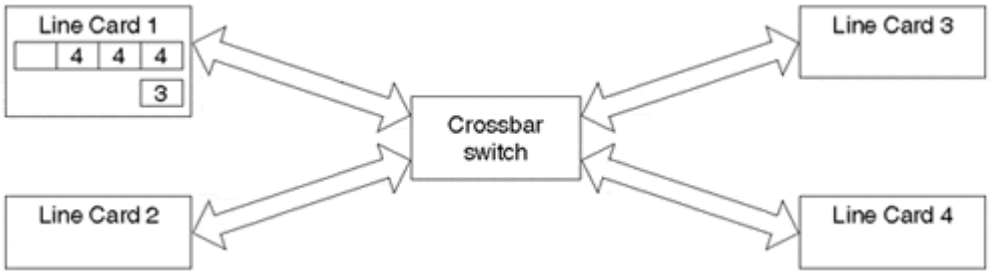


Figure 7-4 shows four different Line Cards connected via a crossbar switch; the packets are labeled according to the LC for which they are destined. The one packet destined to LC 3 cannot be transmitted until the three packets destined to LC 4 are switched through the crossbar switch. This phenomenon, where packets or cells can be held up by packets or cells ahead of them that are destined for a different output, is called Head of Line Blocking.

The GSR solves the HoLB problem by using *virtual output queues*, as shown in Figure 7-5.

Figure 7-5. Virtual Output Queues



Packets or cells destined to each Line Card are placed in a separate queue called the the *virtual output queue*, which eliminates the HoLB problem. These queues are FIFO by default but Modified Deficit Round Robin (MDRR) and Weighted Random Early Detection (WRED) both can be used on these queues to provide quality of service across the Cisco 12000's switching fabric; these QoS mechanisms are covered in more detail in Chapter 8, "Quality of Service."

Cisco Cells

The unit of transfer across the crossbar switch fabric is always fixed size packets, also referred to as Cisco cells,which are easier to schedule than variable size packets. Packets are broken into cells before being placed on the fabric, and are reassembled by the outbound LC before they are transmitted. Cisco cells are 64 bytes long, with an 8-byte header, a 48-byte payload, and an 8-byte CRC.

Maintenance Bus

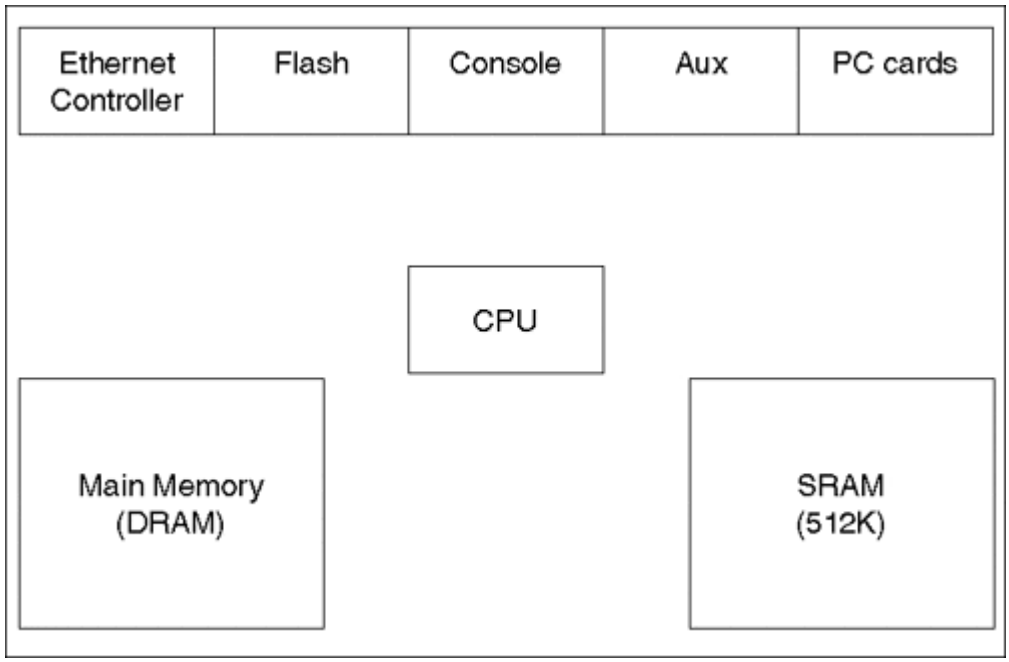
The Maintenance Bus (MBUS) is a 1-Mbps serial bus that connects the Route Processor, the Line Cards, the switch fabric cards, the power supplies, and the fans. The MBUS, as the name suggests, is used for maintenance of the system. When the router boots, the Route Processor uses the MBUS to discover other cards in the system; to boot the Line Cards; and to gather revision numbers, environmental information, and general maintenance information. Data traffic never traverses the MBUS—data traffic always goes across the switching fabric and the MBUS is purely for managing components within the box.

Gigabit Route Processor

The Gigabit Route Processor, more commonly called the GRP, is the brain of the system. The GRP runs routing protocols, computes the forwarding table, builds the CEF and adjacency tables, and distributes them

to all the Line Cards in the system via the switch fabric. Figure 7-6 shows the important components on the GRP, as described in the following list:

Figure 7-6. Route Processor



- **CPU—**

  The CPU on the GRP is the same R5000 processor used on the Cisco 7500's RSP4. The CPU is responsible primarily for running routing protocols and for maintaining a master copy of the CEF table for download to the Line Cards, which do the packet switching.

- **Main Memory (DRAM)—**

  Up to 256 MB used for storing IOS code and all data structures.

- **CSAR SRAM—**

  512 KB; this memory is used for reassembling cells arriving from the switching fabric into packets.

- **Ethernet Controller—**

  Designed for out-of-band management; traffic should not be switched between this port and ports on the LCs.

Line Card

Most packet switching occurs on Line Cards (LCs) in a Cisco GSR. The Line Card architecture can be broken into three sections:

- **Physical layer interface module (PLIM)—**

  The PLIM is the media-specific hardware module (ATM, POS, Fast Ethernet) that terminates the physical connection.

- **Switch Fabric Interface—**

The Switch Fabric Interface prepares packets for transmission across the switching fabric to the destination LC.

- **Packet forwarding engine—**

  The packet forwarding engine switches packets. As of this writing, three types of packet forwarding engines exist; namely, engines 0, 1, and 2. Line Cards available as of this writing are classified by the packet forwarding engine type described in Table 7-3. Engine 0 was the first packet switching engine on the early GSR LCs, engine 1 LCs have some enhancements to improve packet switching performance, and engine 2 LCs further enhance packet forwarding performance and enable QoS features in hardware.

Table 7-3. Classification of GSR Line Cards by Engine

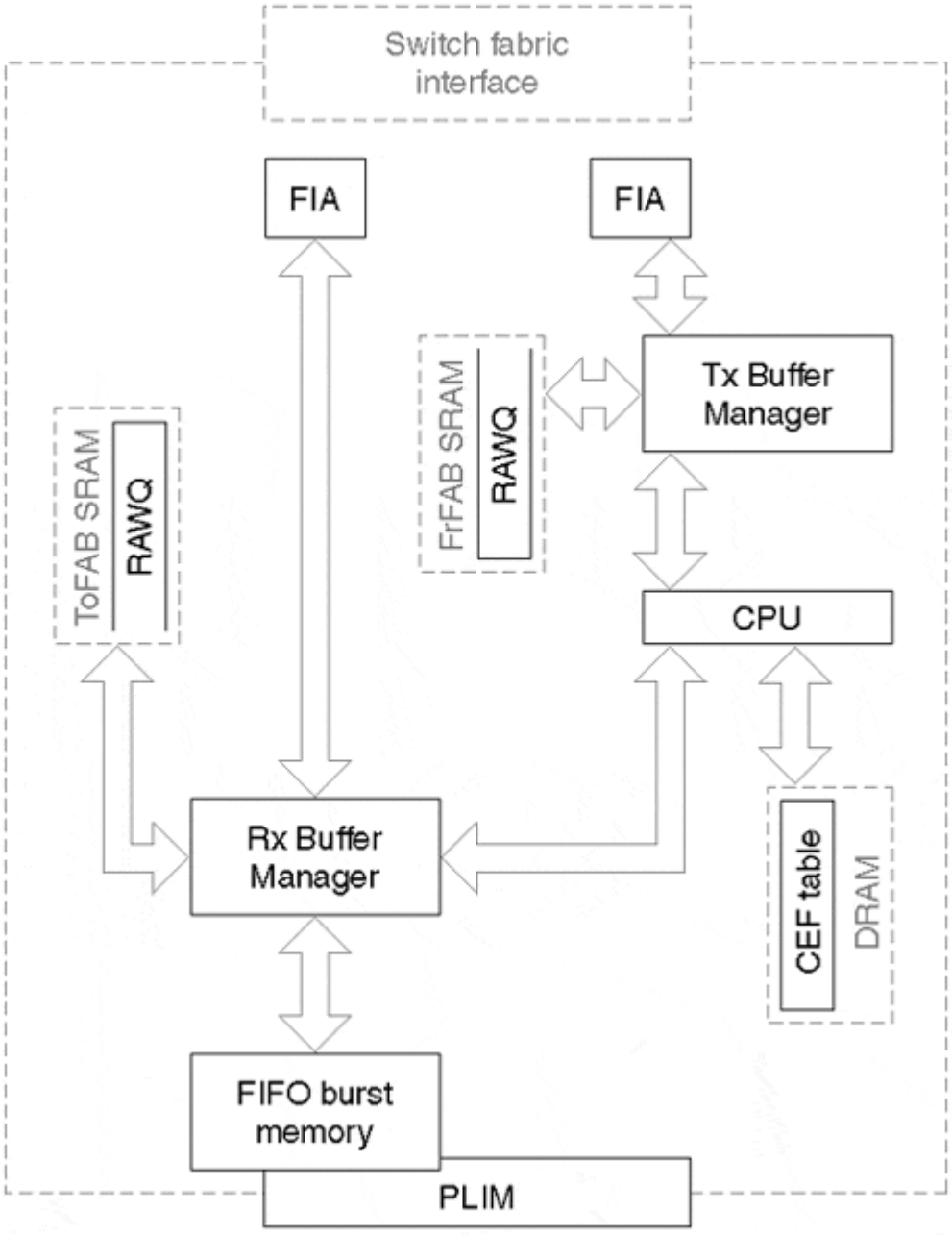
GSR Line Card	Engine 0	Engine 1	Engine 2
OC-12 PoS			
4xOC-3 PoS	X		
Channelized OC-12 to DS-3	X		
OC-12 ATM	X		
6- and 12-Port DS-3	X		
Time to Market (TTM) OC-48 PoS		X	
TTM 4 x OC-12 PoS		X	
1 x Gigabit (GE)		X	
8 x Fast Ethernet (FE)		X	
1 x OC-12 DPT		X	
Enhanced OC-48 PoS			X
Enhanced 4 x OC-12 PoS			X

The first two sections of the LC architecture are self-explanatory. The packet forwarding engine is the heart of the LC, and discussed in detail in the following section.

Packet Forwarding Engines

Three types of packet forwarding engines are available (engines 0, 1, and 2) as of this writing and a detailed description of each follows. Figure 7-7 illustrates the key components of the engine 0 and the engine 1 Line Cards.

Figure 7-7. Engine 0 and Engine 1 Packet Forwarding Engine



Elements common to all engines include main memory and packet memory (receive and transmit), as described in the following sections.

Main Memory

Up to 265 MB of DRAM is used for storing the LCs IOS code and associated data structures, such as the CEF and Adjacency tables.

Packet Memory

Each LC can have up to 256 MB of SRAM (128 MB on engine 0 LCs), equally divided into receive and transmit packet memory. The output of **show diag** displays the amount of receive and transmit packet memory on the LC, as demonstrated in Example 7-2.

Example 7-2. show diag Displays the Amount of Receive and Transmit Packet Memory

Router**#show diag 1** SLOT 1 (RP/LC 1): 4 Port Packet Over SONET OC-3c/STM-1 Multi Mode …. FrFab

SDRAM size: 67108864 bytes ToFab SDRAM size: 67108864 bytes ….

The following list describes what Example 7-2 reveals about transmit and receive packet memory:

- **FrFab SDRAM—**

  Transmit packet memory.

- **ToFab SDRAM—**

  Receive packet memory.

Receive Packet Memory

Receive packet memory is carved into pools of packet buffers. To see how the receive memory is carved, **attach** to a Line Card and execute **show controller tofab queue**, as demonstrated in Example 7-3.

Example 7-3. show controller tofab queue Displays How Receive Packet Memory Is Carved into Packet Buffer Pools

LC-Slot1#**show controller tofab queue** Carve information for ToFab buffers SDRAM size: 67108864 bytes, address: 30000000, carve base: 30019100 67006208 bytes carve size, 0 SDRAM bank(s), 0 bytes SDRAM pagesize, 3 carve(s) max buffer data size 4544 bytes, min buffer data size 80 bytes 65534/65534 buffers specified/carved 66789056/66789056 bytes sum buffer sizes specified/carved Qnum Head Tail #Qelem LenThresh ---- ---- ---- ------ --------- 4 non-IPC free queues: 26174/26174 (buffers specified/carved), 39.93%, 80 byte data size 1 101 26274 26174 65535 19630/19630 (buffers specified/carved), 29.95%, 608 byte data size 2 26275 45904 19630 65535 13087/13087 (buffers specified/carved), 19.96%, 1568 byte data size 3 45905 58991 13087 65535 6543/6543 (buffers specified/carved), 9.98%, 4544 byte data size 4 58992 65534 6543 65535 IPC Queue: 100/100 (buffers specified/carved), 0.15%, 4112 byte data size 30 7 6 100 65535 Raw Queue: 31 0 0 0 65535 ToFab Queues: Dest Slot 0 0 0 0 65535 1 0 0 0 65535 2 0 0 0 65535 3 0 0 0 65535 4 0 0 0 65535 5 0 0 0 65535 6 0 0 0 65535 7 0 0 0 65535 8 0 0 0 65535 9 0 0 0 65535 10 0 0 0 65535 11 0 10 0 65535 12 0 0 0 65535 13 0 0 0 65535 14 0 0 0 65535 15 0 0 0 65535 Multicast 0 0 0 65535

The following list describes some of the key fields in Example 7-3:

- **SDRAM size: 67108864 bytes, address: 30000000, carve base: 30019100—**

  The size of receive packet memory and the address location where it begins.

- **max buffer data size 4544 bytes, min buffer data size 80 bytes—**

  The maximum and minimum buffer sizes.

- **65534/65534 buffers specified/carved—**

  Buffers specified by IOS to be carved and the number of buffers actually carved.

- **4 non-IPC free queues—**

  The non-IPC buffer pools are the packet buffer pools. Packets arriving into the Line Card would be allocated a buffer from one of these buffer pools depending on the size of the packet. The example output shows four packet buffer pools of sizes 80, 608, 1568, and 4544 bytes. For each pool, more detail is given:

  - **26174/26174 (buffers specified/carved), 39.93%, 80 byte data—**

    39.93 percent of the receive packet memory has been carved into 26,174 80-byte buffers.

- **Qnum—**

  The queue number.

- **#Qelem—**

  Number of buffers in this queue.

- **IPC Queue—**

  Reserved for inter-process communication, such as messages from the RP to the Line Cards.

- **Raw Queue—**

  When an incoming packet has been assigned a buffer from a non-IPC free queue, it's enqueued on the raw queue. The raw queue is a FIFO processed by the LC CPU during interrupts.

- **ToFab Queues—**

  Virtual output queues; one per destination slot plus one for multicast traffic. The highlighted portion of Example 7-3, taken from a 12012, shows 15 virtual output queues. The 12012 was designed originally as a 15-slot chassis; queues 13 through 15 are not used.

After the ingress Line Card CPU makes a packet switching decision, the packet is enqueued on the virtual output queue corresponding to the slot where the packet is destined. The number in the fourth column is the number of packets currently enqueued on a virtual output queue.

Transmit Packet Memory

Transmit packet memory is carved into pools as well; attaching to an LC and executing **show controller frfab queue** displays a snapshot of the transmit packet memory. The only field that is different from the output in Example 7-3 is the **Interface Queues** column—the fourth column in the following code.

```
0       0       0               0       65535
1       0       0               0       65535
2       0       0               0       65535
3       0       0               0       65535
```

One interface queue exists for each interface on the LC; packets destined out a specific interface are enqueued onto its interface queue. Now that you have looked at elements common to all engine LCs, take a look at engine-specific details.

Engine 0 Line Card

Each engine 0 LC has two Buffer Manager ASICs (BMAs), represented as Rx and Tx Buffer manager in Figure 7-7, which manage packet buffers and segment and reassemble packets for transmission across the switching fabric, as shown in Figure 7-7. The receive BMA is responsible for receiving packets from the PLIM, segmentation of packets into fixed size cells, and presenting them to the Fabric Interface ASIC (FIA) for transmission across the switch fabric. The transmit BMA, with help from the FIA, performs the re-assembly of the cells arriving from the switch fabric into packets, and hands packets to the PLIM for transmission from the box.

The primary responsibility of the CPU on an engine 0 LC is to switch packets.

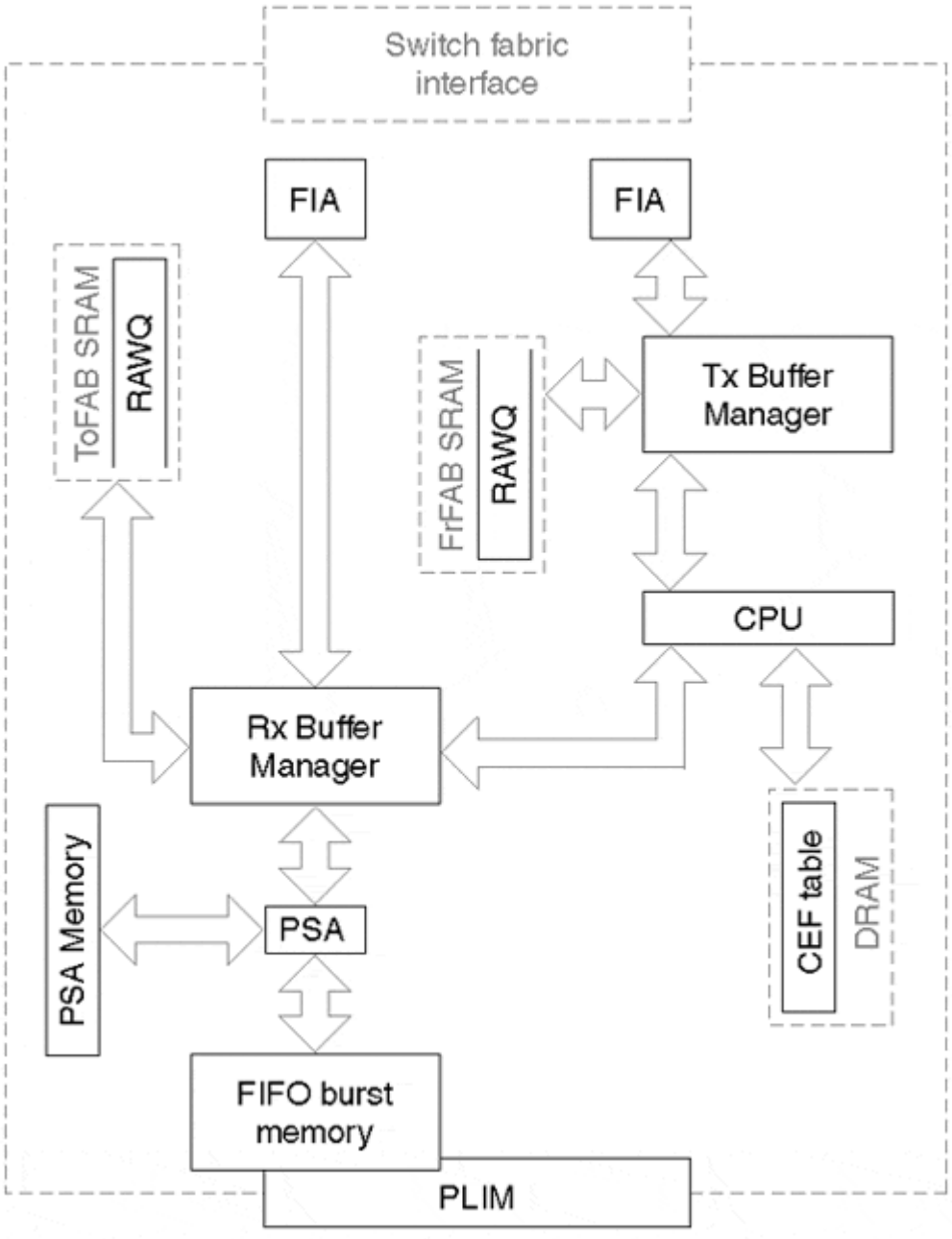
Engine 1 Line Card

Elements common to engine 0 and engine 1 LCs, namely CPU, main memory, and packet memory, have the same function on the engine 1 LC as on the engine 0 LC. Packet switching performance of engine 1 LC is improved from an engine 0 LC by a new buffer manager ASIC and an improved switching engine.

Engine 2 Line Card

Figure 7-8 illustrates the key components of the engine 2 LC. Main memory and packet memory on this LC have the same function as the engine 0 and engine 1 LCs.

Figure 7-8. Engine 2 Packet Forwarding Engine



Engine 2 introduces a new packet switching ASIC (PSA) that greatly improves packet switching performance. A local copy of the global CEF table is stored in the PSA memory (up to 128 MB), which is used for CEF lookup for packet switching. All packet switching on an engine 2 LC is done in hardware by the PSA and the CPU on the LC is never interrupted for any packet forwarding decision. The introduction of new and optimized receive and transmit buffer management ASICs, referred to as Receive Buffer Manager (RBM) and Transmit Buffer Manager (TBM), also is key for hardware-based support for Class of Service (CoS) features on this LC.

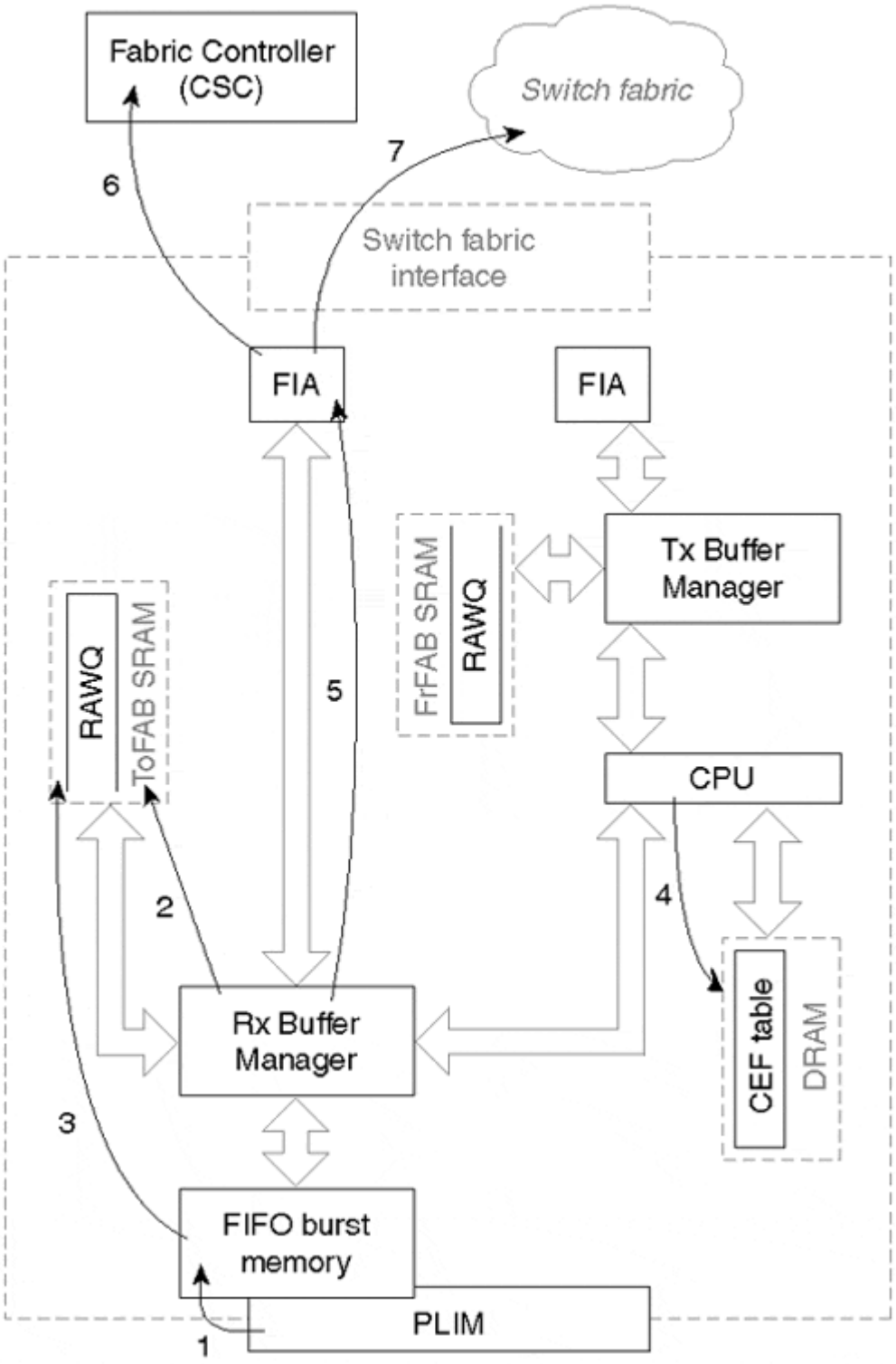
Packet Switching

Now that you've looked at the various components that make up a Cisco 12000, take a look at how they are used to actually switch packets. Most packets switched through a Cisco 12000 are switched by the Line Cards using dCEF; only control packets, such as those used for routing protocols, are punted to the GRP for processing. The packet switching path on the GSR depends on the type of forwarding engine on the LC. Let's look at the switching path with each of the forwarding engines.

Packet Switching: Engine 0 and Engine 1

Figure 7-9 illustrates the steps in the switching packets arriving into an engine 0 or engine 1 LC.

Figure 7-9. Engine 0 and Engine 1 Switching Path



The switching path for engine 0 and engine 1 LCs is essentially the same, although engine 1 LCs have an enhanced switching engine and buffer manager for increased performance. The switching path is as follows:

Step 1. The interface processor (PLIM) detects a packet on the network media and begins copying it into a FIFO memory called *burst memory* on the LC. The amount of burst memory each interface has depends on the type of LC; typical LCs have 128 KB to 1 MB of burst memory.

Step 2. The interface processor requests a packet buffer from the receive BMA; the pool from which the buffer is requested depends on the length of the packet.

If there aren't any free buffers in the requested buffer pool, the packet is dropped and the interface's *ignore* counter is incremented. For example, if a 64-byte packet arrives into an interface, the BMA tries to allocate an 80-byte packet buffer. If no free buffers exist in the 80-byte pool, buffers are not allocated from the next available pool.

Step 3. When a free buffer is allocated by the BMA, the packet is copied into the buffer and is enqueued on the raw queue (RawQ) for processing by the CPU and an interrupt is sent to the LC CPU.

Step 4. The LC's CPU processes each packet in the RawQ as it is received (the RawQ is a FIFO), consulting the local dCEF table in DRAM to make a switching decision.

Step 4.1. If this is a unicast IP packet with a valid destination address in the CEF table, the packet header is rewritten with the new encapsulation information obtained from the CEF adjacency table. The switched packet is enqueued on the virtual output queue corresponding to the destination slot.

Step 4.2. If the destination address is not in the CEF table, the packet is dropped.

Step 4.3. If the packet is a control packet (a routing update, for example), the packet is enqueued on the virtual output queue of the GRP and processed by the GRP.

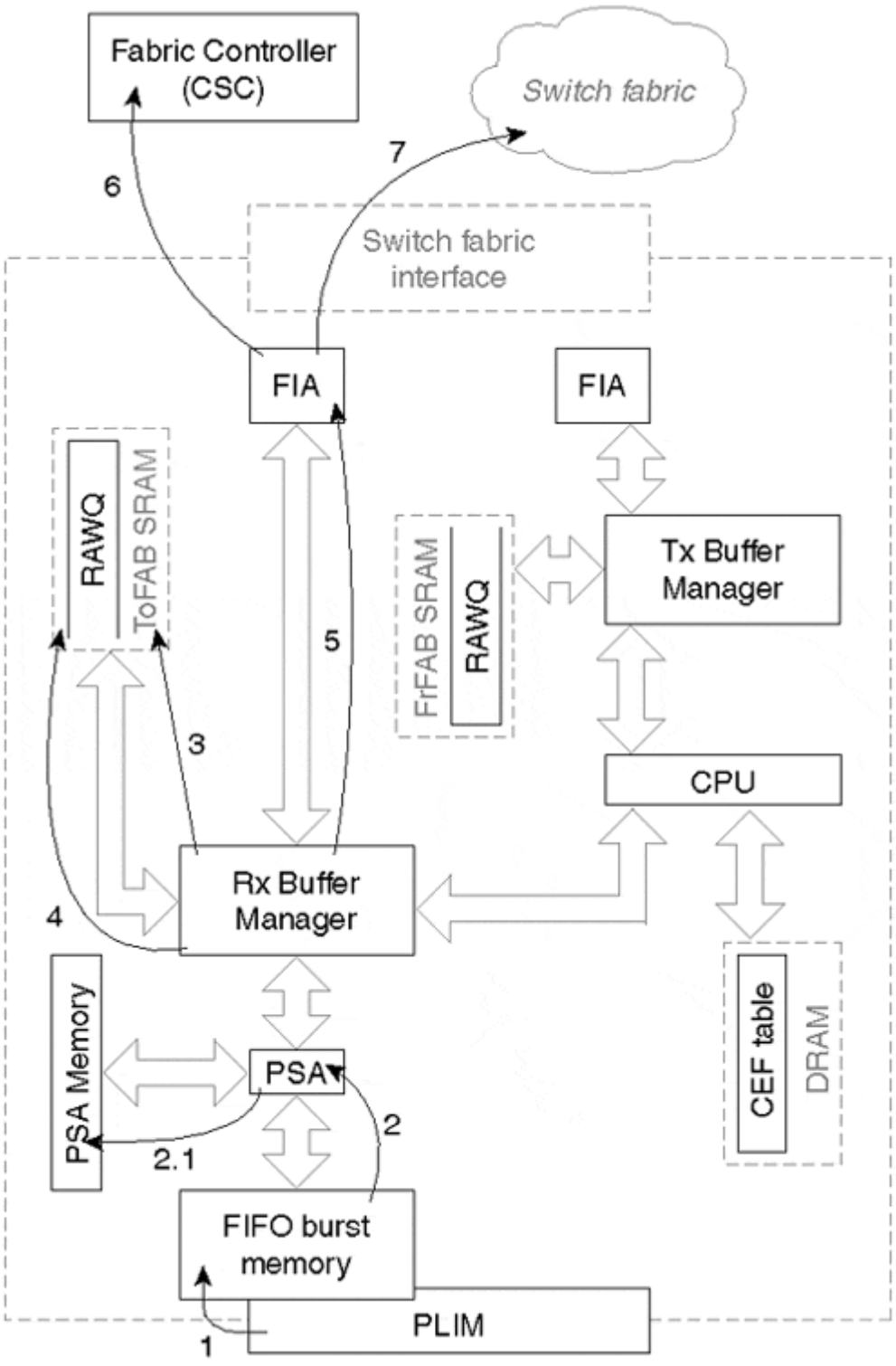
Step 5. The Receive BMA fragments the packet into 64-byte Cisco cells, and hands off these to the FIA for transmission to the outbound LC.

At the end of Step 5, the packet that arrived into an engine 0 LC has been switched and is ready to be transported across the switch fabric as cells. Go to Step 6 described in the section, "Packet Switching: Switching Cells across Fabric."

Packet Switching: Engine 2 LC

Figure 7-10 illustrates the packet switching path when the packets arrive into an engine 2 LC, as described in the following list.

Figure 7-10. Engine 2 Packet Switching Path



Step 1. The interface processor (PLIM) detects a packet on the network media and begins copying it into a FIFO memory called *burst memory* on the LC. The amount of burst memory each interface has depends on the type of LC; typical LCs have 128 KB to 1 MB of burst memory.

Step 2. The first 64 bytes of the packet, called the *header*, is passed through the Packet Switching ASIC (PSA).

Step 2.1. The PSA switches the packet by consulting the local CEF table in the PSA memory. If the packet cannot be switched by the PSA, go to Step 4; otherwise, go to Step 3.

Step 3. The Receive Buffer Manager (RBM) accepts the header from the PSA and copies it into a free buffer from the packet memory based on the length of the packet in the buffer header. If the packet is larger than 64 bytes, the *tail* of the packet also is copied into the same free buffer in packet memory and is queued on the outgoing LC virtual output queue. Go to Step 5.

Step 4. The packet arrives at this step if it cannot be switched by the PSA. These packets are placed on the RawQ and the switching path is essentially the same as engine 1 and engine 0 LC from this point (Step 4 in the case of engine 0). Note that packets that are switched by the PSA are never placed in the RawQ and no interrupt is sent to the CPU.

Step 5. The Fabric Interface Module (FIM) is responsible for segmenting the packets into Cisco cells and sending the cells to the Fabric Interface ASIC (FIA) for transmission to the outbound LC.

Packet Switching: Switching Cells across Fabric

You arrive at this stage after the packet switching engine switches the packets and the packets are segmented into Cisco cells and are waiting to be transmitted across the switch fabric. The steps for this stage are as follows:

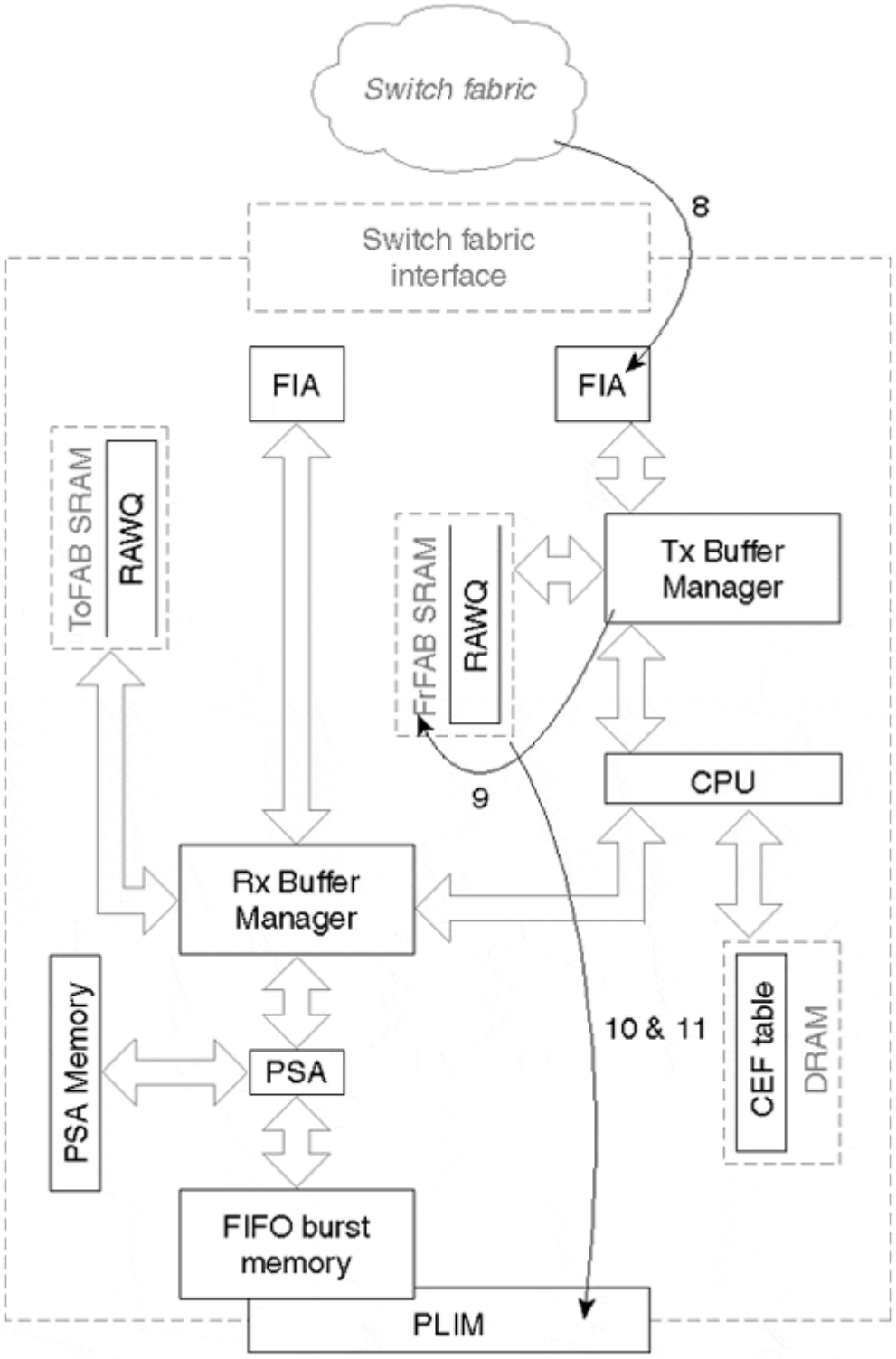
Step 6. The FIA sends a grant request to the CSC, which schedules each cell's transfer across the switch fabric.

Step 7. When the scheduler grants access to the switch fabric, the cells are transferred to the destination slot. Note that the cells might not be transmitted all at once—other cells within other packets might be interleaved.

Packet Switching: Transmitting Packets

Figure 7-11 shows the cells containing the packet on the outbound Line Card.

Figure 7-11. GSR Packet Switching: Transmit Stage



Step 8. The cells switched across the fabric arrive into the destination Line Card via the FIA.

Step 9. The transmit Buffer Manager allocates a buffer from the transmit packet memory and reassembles the packet in this buffer.

Step 10. When the packet is rebuilt, the transmit BMA enqueues the packet onto the destination interface's transmit queue on the LC. If the interface transmit queue is full (the packet cannot be enqueued), the packet is dropped and the *output drop* counter is incremented.

NOTE

In the transmit direction, the only time packets will be placed in the RAWQ is if the LC CPU needs to do any processing before transmission. Examples include IP fragmentation, multicast, and output CAR.

Step 11. The interface processor detects a packet waiting to be transmitted, dequeues the buffer from the transmit memory, copies it into internal FIFO memory, and transmits the packet on the media.

< BACK Make Note | Bookmark CONTINUE >

Index terms contained in this section