Generic Market ACcelerator GMAC

Product Overview

About This Document

This document is the GMAC product overview.

Copyright Information

Celoxica and the Celoxica logo are trademarks of Celoxica Limited.

All other products or services mentioned herein may be trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous development and improvement. All particulars of the product and its use contained in this document are given by Celoxica Limited in good faith. However, all warranties implied or express, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. Celoxica Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any incorrect use of the product. The information contained herein is subject to change without notice and is for general guidance only.

Sales	sales@celoxica.com
Customer Support	support@celoxica.com
Website	http://www.celoxica.com

UK Head Office

Celoxica Limited
34 Porchester Road
London
W2 6ES, UK
Phone: +44 (0) 20 7313 3180

US Head Office

Celoxica Inc.
275 Madison Avenue, Suite 404
New York, NY
10016, USA
Phone: +1 (0) 212 880 2075

US Chicago Office

Celoxica Inc.
141 W Jackson Blvd, Suite 2350
Chicago, IL
60604, USA
Phone: +1 (0) 312 893 1204

Content

Table of Figures

Revisions

Revision	Date	Description of Changes
R2012-8.1	20 DEC 2012	Release R2012-8.1 - Revised Terminology section
1.02	19 SEP 2012	16 DMA queues are supported by both 1G and 10G cards
1.01	29 AUG 2012	Update of the section describing the API variants
1.0	10 FEB 2012	Initial release

1. GMAC Overview

1.1 Introduction

This document should be read in conjunction with:

- GMAC API Reference Guide
- GMAC Configuration Guide
- GMAC V1 and V3 Messages Templates

1.2 Principles

Celoxica's Generic Market ACcelerator GMAC is an FPGA-based solution providing low-latency access to inbound market data feeds and high-performance communication with a client software application.

User applications integrated with GMAC are able to leverage its functionality, which hides the complexities of working with multiple market data formats, handles synchronisation issues transparently, and facilitates order book building and data redistribution.

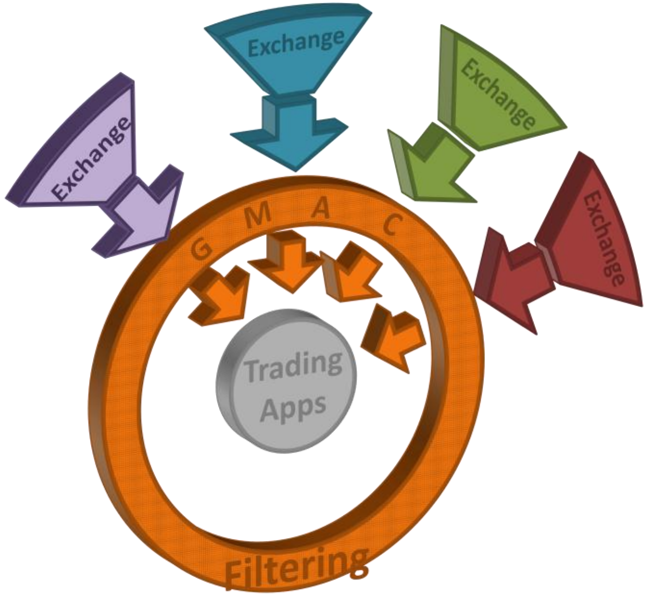

Figure 1 – GMAC in its environment

GMAC offers the following features:

- **Configuration**: The parameterisation of GMAC using a configuration tree;
- **Line Handling**: Handling market data from a physical ethernet connection via the accelerator card;
- **Hardware Arbitrage**: The hardware A/B Redundancy Arbitration feature consists in analyzing two identical feeds nominally labeled A and B, to fill any sequence number gaps and to deliver packets from the fastest of the two feeds;
- **Hardware Symbol Filtering**: Filtering market data according to instrument name and index so that only the required market data feed is transmitted to the GMAC software-side;
- **Re-ordering**: The sorting and re-ordering received market data in order to deliver a coherent stream of messages;

- **Recovery**: The automatic detection of gaps in the inbound market data, and the mechanism to fill those gaps by re-requesting data from the exchange;
- **Snapshot / Refresh**: The mechanism to allow user applications to request snapshots and to resynchronise their books;
- **Normalization / Generic API**: The transcoding of market data from supported market data feeds into a normalized data format and the exposure of a generic API for interfacing to multi-threaded user applications.

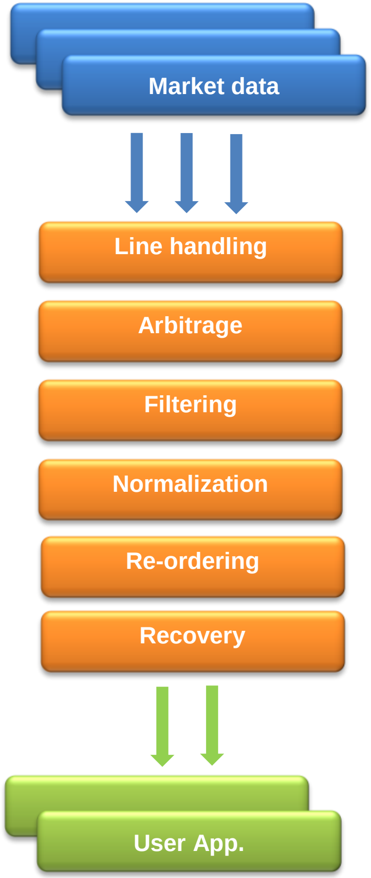

Figure 2 – GMAC functional overview

The feed is received directly on the FPGA through multiple 10 GE ports. The Accelerator Card handles the line data processing and forwards the traffic to the host running the software-side of the GMAC. The software-side connects to the multi-threaded user application through a standard API. The communication between the Accelerator Card and the host is provided by Celoxica’s Low Latency Data Transfer (LLDT) link transparently to the user application.

1.3 Configuration

GMAC is configured using a GMAC configuration file which implements a configuration tree upon GMAC initialization.

The configuration tree allows GMAC users to set two configuration parameters types:

- **Global settings**: verbosity, recovery settings …
- **Market settings**: specific market parameters such as IP addresses for market data channels, credentials for interactive channels and lists of instruments for hardware filtering.

More details are provided in the GMAC Configuration Guide.

1.4 Data Flow

1.4.1 Line Handling

Market data suppliers transmit market data on one or more duplicated feeds, with each feed being sent over a dedicated physical line. Each feed consists of data sent on one or more UDP multicasts or one or more TCP/IP channels. Multicast channels for each feed are generally transmitted from a single source IP address and port.

GMAC routes these multicast channels to market logical GMAC channels for processing. Processing of a channel encompasses arbitrage, filtering, re-ordering/recovery, normalization and delivery to the user process, which runs on the same CPU being used to poll data from the accelerator card. The CPU receives normalized data which the user application can use to build its Order Book.

Channels can be routed to DMA queues. Each DMA queue delivers normalized data to a single predetermined server CPU; this allows efficient work balancing across multiple CPUs in the server. DMA queues are not restricted to single markets; this means that channels from one or more markets can be grouped together for delivery to a single server CPU.

The following diagram illustrates the data flow through GMAC.

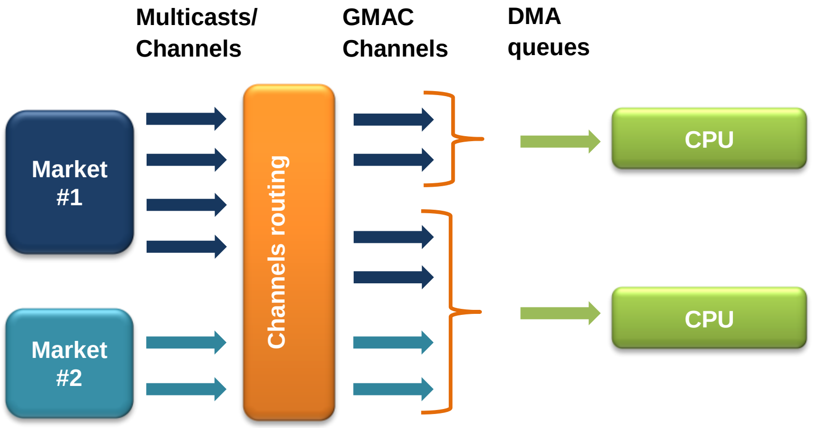

Figure 3 – Line handling through GMAC

The number of CPUs is equivalent to the number of DMA queues since one CPU is pegged to each DMA Queue.

The number of user-defined groups of market logical channels is restricted by the number of DMA queues supported by the accelerator card(s) in the host system.

In current firmware, 16 DMA queues are supported for both 1G and 10G cards.

1.4.2 Hardware Symbol Filtering

This feature consists in filtering market data according to symbol name and index, so that only market data for specific symbols are processed, with the filtering taking place in the accelerator card. This leads to improved overall performance since ‘non-interesting’ data are dropped by the card, so are never normalized or passed to the GMAC software-side.

The firmware for 10G cards supports both filtering and non-filtering. By default the filtering logic is bypassed.

Hardware symbol filtering is controlled via the GMAC API, and enabled via the GMAC Configuration file – see the GMAC API Reference Guide and GMAC Configuration Guide for details.

Hardware instrument-filtering can be enabled and disabled per market.

1.4.3 Normalization

GMAC normalizes market data from all supported exchanges into a single data format. GMAC can generate 4 types of messages:

- **Order book updates**: These messages contain all the necessary information to build the Order Book. These can miss some information, for example currency or price factor;
- **Data referential**: These messages contain the information not included in the order updates. For example currency or price factor;
- **Trading status updates**: Delivers the trading status on instruments, segments, channels or markets;
- **User control messages**: Sent by the GMAC users to allow efficient and easy distribution of control information across multiple CPUs. These messages are received like any other GMAC message on the destination CPU. The call to send this type of message is thread-safe.

1.4.4 Recovery

Since UDP is unreliable by design, various data recovery features have been implemented within GMAC.

1.4.4.1 Hardware Line Arbitration

UDP IP protocol market feeds are often transmitted on duplicate physical lines, denoted A and B. The A and B data streams are theoretically identical and originate from different hardware, and thus have different source IPs. The A and B feeds may be used together to mitigate the risk of receiving gaps in the market data, which can happen since UDP is unreliable by design.

Celoxica's accelerator hardware is able to perform this function, which is referred to as line arbitration. Filter modules in the hardware split the IP data stream into the constituent groups. Corresponding 'A' and 'B' streams are then paired together before being fed into the arbitration core.

The arbitrage core takes pairs of streams and performs arbitration by storing the presence of a packet indexed by its sequence number and comparing it to the presence of the same packet for the other feed. In this way, each feed complements the other and duplicate packets (i.e. packets in either feed that have already been seen) are dropped.

Note:

The output from the arbitration core gives priority to feed A. That is, if at a given time a given sequence number is present in both feeds, the packet from the 'A' feed will be used and the packet from the 'B' feed will be dropped.

Packets from the resulting stream are then processed by the GMAC libraries, where any gaps that could not be filled by the arbitrage module, are detected. If gap recovery is enabled in GMAC and supported for the feed in question, the missing packets will be re-requested. Otherwise GMAC passes the gap to the user, who can decide on the appropriate course of action.

Feeding only feed A or feed B through redundancy arbitration will result in the original feed being transferred and no arbitrage being performed.

Hardware Line arbitration is configured via the GMAC configuration file – see the GMAC Configuration Guide for details.

Below is an example showing the effect of the A/B Redundancy Arbitration feature. The numbers shown are the sequence numbers of the packets.

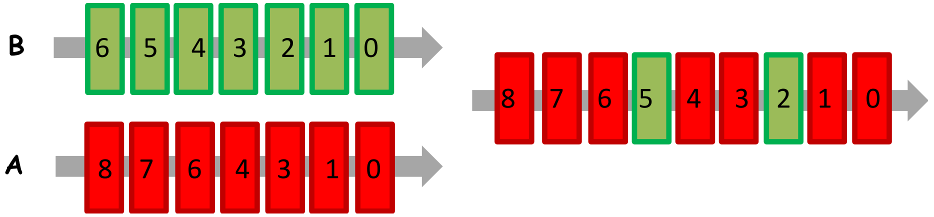

Figure 4 – Packets line arbitration

See the GMAC Market Support Matrix for details on feeds supporting the hardware line arbitration feature.

1.4.4.2 Re-ordering / Recovery

Re-ordering is performed when recovery is enabled. This feature consists in re-ordering the packets once dropped packets have been recovered.

There are two types of recovery:

- **Gap recovery**: recovery of a limited number of dropped packets;
- **Snapshot recovery**: snapshot of current market state.

Recovery is configured per channel in the GMAC configuration file.

1.4.4.3 Gap Recovery

Many feeds support gap recovery. Gap recovery enables market subscribers to request retransmission of market data with specific sequence numbers when packets have been dropped on both line A and line B. This is usually done using a TCP/IP connection.

When a missing sequence number is detected by GMAC, the missing sequence number is requested and incoming data is buffered until the sequence number is received. On some feeds a timeout can be configured for this waiting period in the GMAC configuration file.

See the GMAC Market Support Matrix for details on feeds supporting the gap recovery feature.

1.4.4.4 Snapshot Recovery

Some feeds support snapshot/refresh recovery. These feeds are able to transmit complete snapshots of the Order Book contents. This can be used to check the state of the order book or to build the entire order book at a convenient time, for example during a midday-start or following an outage.

The snapshot recovery is the first step to get back in synchronization with the real time market.

Some feeds support snapshot requests and others transmit snapshots periodically, GMAC supports both of these types.

See the GMAC Market Support Matrix for details on feeds supporting the snapshot recovery feature.

1.5 API

GMAC ships as a set of dynamic libraries, which the user code can interface to via the API it exposes. Currently GMAC has the following variants of the API:

- V3: The new API, supporting a generic and logically extensible interface. This variant is the new standard for all plugins;
- V3C: The consolidated version of the V3 API, for consolidated data feeds such as OPRA, CQS, etc.
- V1: Original API variant. It is deprecated but still supported.

All API variants enable multi-threaded integration of GMAC into user code.

See the GMAC Market Support Matrix for details of which version of the API to use with each supported feed.

See the GMAC API Reference Guide for details about the API functions.

1.5.1 User's Application

Initializing GMAC in a user's application requires only a few GMAC API function calls. The `GMACOpen()` API function call initializes the GMAC handle and configuration node. The configuration node contains all the GMAC configuration parameters which are currently set in the GMAC configuration file.

The user application then opens the DMA queues configured in the GMAC configuration file. This is done with a `GMACQueueOpen()` API function call for each DMA queue. `GMACQueueOpen()` API function calls should be made in parallel, enabling multiple CPUs to be used for work balancing.

When the user wants to stop receiving data, the DMA queues have to be closed immediately before closing the GMAC handle.

1.5.2 Market Data Messages

GMAC market data messages are grouped into three distinct categories:

- **Market data generic messages**: these messages are generated as normalized messages for various exchange data feeds;
- **Consolidated generic messages**: these messages are generated as normalized messages for the American consolidated data feeds (OPRA, CQS, CTS, UQDF, UTDF);
- **Specific messages**: these messages are exchange specific messages.

The user's application reads the DMA queues via `GMACQueueRead()` API function to continuously retrieve the normalized messages.

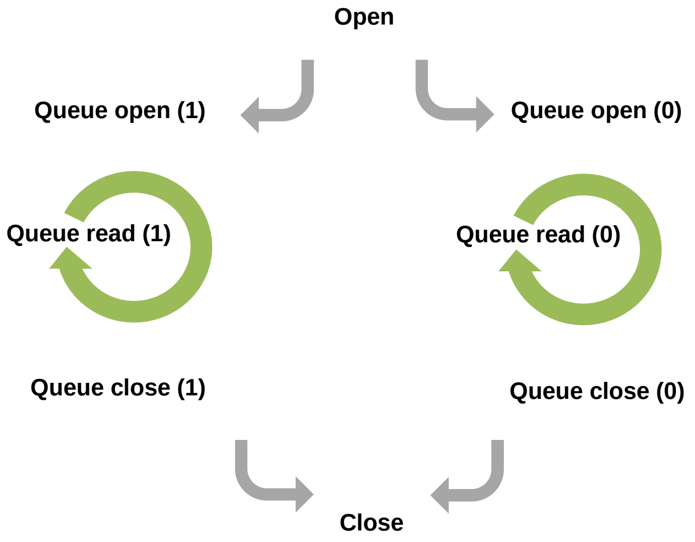

Figure 5 – API function usage

1.5.3 Latency Measurement

GMAC offers a true wire to software-user-defined-point averaged latency measurement using latency API functions. Latency measurement is very light-weight and should not influence performance.

1.5.4 Statistics

The user’s application can report statistics at any time by calling the statistics API function to get the number of delivered packets, received packets, gaps detected …

1.6 Physical Architecture Requirements

See the PCI Express Card Software Installation Guide for details.

2. Terminology

ADE Accelerated Data Engine. The ADE is a library which controls the inputs and outputs of market data on systems fitted with Celoxica hardware.

Arbitrage A/B Redundancy Arbitration consists in analyzing two identical feeds nominally labeled ‘A’ and ‘B’ to fill any sequence number gaps and to deliver packets from the fastest of the two feeds.

DMA Queue A group of channels delivered from GMAC directly to a single CPU for immediate processing in a user’s application. Channels may have data from several different markets.

Exchange A stock exchange, a derivatives exchange, a FX platform or a cash bond platform; transmits data and receives order transactions on one or more markets.

Feed Market data transmitted on a single physical line (Ethernet connection). Feeds consist of one or more multicasts.

GMAC Generic Market ACcelerator, combined hardware and software solution which receives market data, reorders and normalizes it and delivers it directly to the CPU for processing in the user’s application.

GMAC Channel A logically separated stream of data within GMAC. A GMAC channel contains one or more multicasts from one or more markets which are reordered, normalized and directed to a DMA queue.

Hardware The hardware refers to the Celoxica’s FPGA-based Accelerator Card.

Hardware Symbol Filtering Instrument-filtering market data so that only the required market data is transmitted to the GMAC software-side.

Market Transmits data on one or more duplicated feeds and receives order transactions via TCP connections.

Multicast Channel Market data transmitted from a single source IP address and port.

Normalization Normalization is the process performed by GMAC which converts market data from all supported exchanges into a single (normalized) data format. This means applications written using the GMAC API work for all exchanges.

Recovery Process of recovering of dropped packets.

Re-ordering Process of re-ordering a data flow when dropped packets have been recovered.