

9061906  
17611  
U.S. Patent and Trademark Office

PTO/SB/16 (10-05)

Approved for use through 07/31/2006. OMB 0651-0032

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE


Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

# **PROVISIONAL APPLICATION FOR PATENT COVER SHEET – Page 1 of 2**

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Express Mail Label No.

**EV 387025868 US**

INVENTOR(S)					
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)			
Roger D.	Chamberlain	64 Notre Dame Drive St. Louis, Missouri 63141			
Scott	Parsons	10 Beaufort Court St. Charles, MO 63301			
David E.	Taylor	3448 Missouri Avenue St. Louis, Missouri 63118			
Rod	Arbaugh	232 N. Kingshighway #1203 St. Louis, Missouri 63108			
Mark A.	Franklin	7456 Stratford Avenue, St. Louis, Missouri 63130			
Additional inventors are being named on the <u>0</u> separately numbered sheets attached hereto					
TITLE OF THE INVENTION (500 characters max):					
HIGH SPEED PROCESSING OF FINANCIAL INFORMATION USING FPGA DEVICES					
Direct all correspondence to: CORRESPONDENCE ADDRESS					
<input checked="" type="checkbox"/> The address corresponding to Customer Number: <div style="border: 1px solid black; padding: 5px; display: inline-block;">   21888 </div>					
<input type="checkbox"/> Firm or Individual Name Benjamin L. Volk, Jr. Address Thompson Coburn LLP One US Bank Plaza City St. Louis State MO Zip 63101 Country U.S.A. Telephone 314-552-6000 Fax 314-552-7000					
ENCLOSED APPLICATION PARTS (check all that apply)					
<input type="checkbox"/> Application Data Sheet. See 37 CFR 1.76		<input type="checkbox"/> CD(s), Number of CDs _____			
<input checked="" type="checkbox"/> Specification Number of Pages <u>35</u>		<input checked="" type="checkbox"/> Other (specify) <u>Postcard</u>			
<input checked="" type="checkbox"/> Drawing(s) Number of Sheets <u>9</u>					
<b>Fees Due:</b> Filing Fee of \$200 (\$100 for small entity). If the specification and drawings exceed 100 sheets of paper, an application size fee is also due, which is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).					
METHOD OF PAYMENT OF THE FILING FEES AND APPLICATION SIZE FEE FOR THIS PROVISIONAL APPLICATION FOR PATENT					
<input checked="" type="checkbox"/> Applicant claims small entity status. See 37 CFR 1.27.		<div style="border: 1px solid black; padding: 5px; display: inline-block;">\$100.00</div> <b>TOTAL FEE AMOUNT (\$)</b>			
<input type="checkbox"/> A check or money order is enclosed to cover the filing fee and application fee (if applicable).					
<input type="checkbox"/> Payment by credit card. Form PTO-2038 is attached.					
<input checked="" type="checkbox"/> The Director is hereby authorized to charge filing fee and application size fee (if applicable) or credit any overpayment to Deposit Account Number: <u>20-0823</u> .		A duplicative copy of this form is enclosed for fee processing.			

113374 U.S. PTO  
60/814796  
9061906

**USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT**

This collection of information is required by 37 CFR 1.51. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

3374357

**PROVISIONAL APPLICATION COVER SHEET**  
**Page 2 of 2**

PTO/SB/16 (10-05)

Approved for use through 07/31/2006. OMB 0651-0032

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

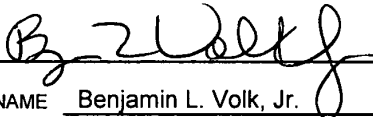
☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are: \_\_\_\_\_

**WARNING:**

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

SIGNATURE



Date

June 19, 2006

TYPED or PRINTED NAME

Benjamin L. Volk, Jr.

REGISTRATION NO.

#48,017

(if appropriate)

TELEPHONE

314-552-6000

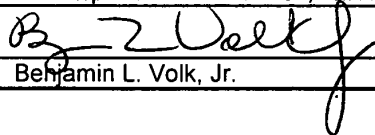
Docket Number:

44826/62269

**CERTIFICATE OF EXPRESS MAILING**

I certify that this document and fee is being deposited with the United States Postal Service as "Express Mail" under 37 CFR 1.10 on June 19, 2006, and addressed to: Mail Stop PATENT APPLICATION, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Signature



Typed or printed name

Benjamin L. Volk, Jr.

5      High Speed Processing of Financial Information Using FPGA DevicesCross-Reference to Related Patent Applications:

        This patent application is related to the following patent applications: United States patent application 09/545,472 (filed  
10      April 7, 2000, and entitled "Associative Database Scanning and Information Retrieval", now United States patent 6,711,558), United States patent application 10/153,151 (filed May 21, 2002, and entitled "Associative Database Scanning and Information Retrieval using FPGA  
15      Devices" and published as 2003/0018630), published PCT applications WO 05/048134 and WO 05/026925 (both filed May 21, 2004, and entitled "Intelligent Data Storage and Processing Using FPGA Devices"), United States provisional patent application 60/658,418 (filed March 3, 2005, and entitled "Biosequence Similarity Searching Using FPGA Devices"),  
20      United States provisional patent application 60/736,081 (filed November 11, 2005, and entitled "Method and Apparatus for Performing Biosequence Similarity Searching"), PCT patent application PCT/US2006/006105 (filed February 22, 2006, and entitled Method and Apparatus for Performing Biosequence Similarity Searching), United States patent application 11/293,619 (filed December 2, 2005, and  
25      entitled "Method and Device for High Performance Regular Expression Pattern Matching"), United States patent application 11/339,892 (filed January 26, 2006, and entitled "Firmware Socket Module for FPGA-Based Pipeline Processing"), and United States patent application 11/381,214 (filed May 2, 2006, and entitled "Method and Apparatus for Approximate  
30      Pattern Matching"), the entire disclosures of each of which are incorporated herein by reference.

Field of the Invention:

        The present invention relates to the field of data processing  
35      platforms for financial market data.

Background and Summary of the Invention:

Speed of information delivery is a valuable dimension to the financial instrument trading and brokerage industry. The ability of a trader to obtain pricing information on financial instruments such as stocks, bonds and particularly options as quickly as possible cannot be understated; improvements in information delivery delay on the order of fractions of a second can provide important value to traders.

For example, suppose there is an outstanding "bid" on stock X that is a firm quote to buy 100 shares of Stock X for \$21.50 per share. Also suppose there are two traders, A and B, each trying to sell 100 shares of stock X, but would prefer not to sell at a price of \$21.50. Next, suppose another party suddenly indicates a willingness to buy 100 shares of Stock X for a price of \$21.60. A new quote for that amount is then submitted, which sets the "best bid" for Stock X to \$21.60, up 10 cents from its previous value of \$21.50. The first trader, A or B, to see the new best bid price for Stock X and issue a counter-party order to sell Stock X will "hit the bid", and sell his/her Stock X for \$21.60 per share. The other trader will either have to settle for selling his/her shares of Stock X for the lower \$21.50 price or will have to decide not to sell at all at that lower price. Thus, it can be seen that speed of information delivery can often translate into actual dollars and cents for traders, which in large volume situations can translate to significant sums of money.

In an attempt to promptly deliver financial information to interested parties such as traders, a variety of market data platforms have been developed for the purpose of ostensible "real time" delivery of streaming bid, offer, and trade information for financial instruments to traders. Figure 1 illustrates an exemplary platform that is currently known in the art. As shown in Figure 1, the market data platform 100 comprises a plurality of functional units 102 that are configured to carry out data processing operations such as the ones depicted in units 102, whereby traders at workstations 104 have access to financial data of interest and whereby trade information can be sent to various exchanges or other outside systems via output path

112. The purpose and details of the functions performed by functional units 102 are well-known in the art. A stream 106 of financial data arrives at the system 100 from an external source such as the exchanges themselves (e.g., NYSE, NASDAQ, etc.) over private data communication lines or from extranet providers such as Savvis or BT Radians. The financial data source stream 106 comprises a series of messages that individually represent a new offer to buy or sell a financial instrument, an indication of a completed sale of a financial instrument, notifications of corrections to previously-reported sales of a financial instrument, administrative messages related to such transactions, and the like. As used herein, a "financial instrument" refers to a contract representing equity ownership, debt or credit, typically in relation to a corporate or governmental entity, wherein the contract is saleable. Examples of "financial instruments" include stocks, bonds, commodities, currency traded on currency markets, etc. but would not include cash or checks in the sense of how those items are used outside financial trading markets (i.e., the purchase of groceries at a grocery store using cash or check would not be covered by the term "financial instrument" as used herein; similarly, the withdrawal of \$100 in cash from an Automatic Teller Machine using a debit card would not be covered by the term "financial instrument" as used herein). Functional units 102 of the system then operate on stream 106 or data derived therefrom to carry out a variety of financial processing tasks. As used herein, the term "financial market data" refers to the data contained in or derived from a series of messages that individually represent a new offer to buy or sell a financial instrument, an indication of a completed sale of a financial instrument, notifications of corrections to previously-reported sales of a financial instrument, administrative messages related to such transactions, and the like. The term "financial market source data" refers to a feed of financial market data received directly from a data source such as an exchange itself or a third party provider (e.g., a Savvis or BT Radianz provider). The term "financial market secondary data" refers to financial market data that has been derived from financial market source data, such as data produced by a feed

compression operation, a feed handling operation, an option pricing operation, etc.

Because of the massive computations required to support such a platform, current implementations known to the inventors herein  
5 typically deploy these functions across a number of individual computer systems that are networked together, to thereby achieve the appropriate processing scale for information delivery to traders with an acceptable degree of latency. This distribution process involves partitioning a given function into multiple logical units and  
10 implementing each logical unit in software on its own computer system/server. The particular partitioning scheme that is used is dependent on the particular function and the nature of the data with which that function works. The inventors believe that a number of different partitioning schemes for market data platforms have been  
15 developed over the years. For large market data platforms, the scale of deployment across multiple computer systems and servers can be physically massive, often filling entire rooms with computer systems and servers, thereby contributing to expensive and complex purchasing, maintenance, and service issues.

20 This partitioning approach is shown by Figure 1 wherein each functional unit 102 can be thought of as its own computer system or server. Buses 108 and 112 can be used to network different functional units 102 together. For many functions, redundancy and scale can be provided by parallel computer systems/servers such as those shown in  
25 connection with options pricing and others. To the inventors' knowledge, these functions are deployed in software that is executed by the conventional general purpose processors (GPPs) resident on the computer systems/servers 102. The nature of general purpose processors and software systems in the current state of the art known  
30 to the inventors herein imposes constraints that limit the performance of these functions. Performance is typically measured as some number of units of computational work that can be performed per unit time on a system (commonly called "throughput"), and the time required to perform each individual unit of computational work from start to  
35 finish (commonly called "latency" or delay). Also, because of the many

physical machines required by system 100, communication latencies are introduced into the data processing operations because of the processing overhead involved in transmitting messages to and from different machines.

5         Despite the improvements to the industry that these systems have provided, the inventors herein believe that significant further improvements can be made. In doing so, the inventors herein disclose that the underlying technology disclosed in the related patent and patent applications listed below can be harnessed to fundamentally  
10         change the system architecture in which market data platforms are deployed; these related patent and patent applications are: United States patent 6,711,558 (entitled "Associative Database Scanning and Information Retrieval"), United States patent application 10/153,151 (filed May 21, 2002, and entitled "Associative Database Scanning and  
15         Information Retrieval using FPGA Devices" and published as 2003/0018630), published PCT applications WO 05/048134 and WO 05/026925 (both filed May 21, 2004, and entitled "Intelligent Data Storage and Processing Using FPGA Devices"), United States provisional patent application 60/658,418 (filed March 3, 2005, and entitled  
20         "Biosequence Similarity Searching Using FPGA Devices"), United States provisional patent application 60/736,081 (filed November 11, 2005, and entitled "Method and Apparatus for Performing Biosequence Similarity Searching"), PCT patent application PCT/US2006/006105 (filed February 22, 2006, and entitled Method and Apparatus for  
25         Performing Biosequence Similarity Searching), United States patent application 11/293,619 (filed December 2, 2005, and entitled "Method and Device for High Performance Regular Expression Pattern Matching"), United States patent application 11/339,892 (filed January 26, 2006, and entitled "Firmware Socket Module for FPGA-Based Pipeline  
30         Processing"), and United States patent application 11/381,214 (filed May 2, 2006, and entitled "Method and Apparatus for Approximate Pattern Matching"), the entire disclosures of each of which are incorporated herein by reference.

       In above-referenced related patent application 10/153,151, it was  
35         first disclosed that reconfigurable logic, such as Field Programmable

Gate Arrays (FPGAs), can be deployed to process streaming financial information at hardware speeds. As examples, the 10/153,151 application disclosed the use of FPGAs to perform data reduction operations on streaming financial information, with specific examples of such data reduction operations being a minimum price function, a maximum price function, and a latest price function.

Since that time, the inventors herein have greatly expanded the scope of functionality for processing streams of financial information with reconfigurable logic. With the invention described herein, vast amounts of streaming financial information can be processed with varying degrees of complexity at hardware speeds via reconfigurable logic deployed in hardware appliances that greatly consolidate the distributed GPP architecture shown in Figure 1 such that a market data platform built in accordance with the principles of the present invention can be implemented within fewer and much smaller appliances while providing faster data processing capabilities relative to the conventional market data platform as illustrated by Figure 1; for example, the inventors envision that a 5:1 or greater reduction of appliances relative to the system architecture of Figure 1 can be achieved in the practice of the present invention.

As used herein, the term "general-purpose processor" (or GPP) refers to a hardware device that fetches instructions and executes those instructions (for example, an Intel Xeon processor or an AMD Opteron processor). The term "reconfigurable logic" refers to any logic technology whose form and function can be significantly altered (i.e., reconfigured) in the field post-manufacture. This is to be contrasted with a GPP, whose function can change post-manufacture, but whose form is fixed at manufacture. The term "software" will refer to data processing functionality that is deployed on a GPP. The term "firmware" will refer to data processing functionality that is deployed on reconfigurable logic.

Thus, as embodiments of the present invention, the inventors herein disclose a variety of data processing pipelines implemented in firmware deployed on reconfigurable logic, wherein a stream of financial data can be processed through these pipelines at hardware

speeds. These and other features and advantages of the present invention will be understood by those having ordinary skill in the art upon review of the description and figures hereinafter.

5 Brief Description of the Drawings:

Figure 1 depicts an exemplary system architecture for a conventional market data platform;

Figure 2 is a block diagram view of an exemplary system architecture in accordance with an embodiment of the present invention;

Figure 3 illustrates an exemplary framework for the deployment of software and firmware for an embodiment of the present invention;

Figure 4(a) is a block diagram view of a preferred printed circuit board for installation into a market data platform to carry out data processing tasks in accordance with the present invention;

Figure 4(b) is a block diagram view of an alternate printed circuit board for installation into a market data platform to carry out data processing tasks in accordance with the present invention;

Figure 5 illustrates an example of how the firmware application modules of a pipeline can be deployed across multiple FPGAs;

Figure 6 illustrates an exemplary architecture for a market data platform in accordance with an embodiment of the present invention;

Figure 7 illustrates an exemplary firmware application module pipeline for a message transformation from the Financial Information Exchange (FIX) format to the FIX Adapted for Streaming (FAST) format;

Figure 8 illustrates an exemplary firmware application module pipeline for a message transformation from the FAST format to the FIX format;

Figure 9 illustrates an exemplary firmware application module pipeline for message format transformation, message data processing, and message encoding; and

Figure 10 illustrates another exemplary firmware application module pipeline for message format transformation, message data processing, and message encoding.

Detailed Description of the Preferred Embodiment:

Figure 2 depicts an exemplary system 200 in accordance with the present invention. In this system, a reconfigurable logic device 202 is positioned to receive data that streams off either or both a disk subsystem defined by disk controller 206 and data store 204 (either  
5 directly or indirectly by way of system memory such as RAM 210) and a network data source/destination 242 (via network interface 240) . Preferably, data streams into the reconfigurable logic device by way of system bus 212, although other design architectures are possible  
10 (see Figure 4(b)). Preferably, the reconfigurable logic device 202 is a FPGA, although this need not be the case. System bus 212 can also interconnect the reconfigurable logic device 202 with the computer system's main processor 208 as well as the computer system's RAM 210. The term "bus" as used herein refers to a logical bus which  
15 encompasses any physical interconnect for which devices and locations are accessed by an address. Examples of buses that could be used in the practice of the present invention include, but are not limited to the PCI family of buses (e.g., PCI-X and PCI-Express) and HyperTransport buses. In a preferred embodiment, system bus 212 may  
20 be a PCI-X bus, although this need not be the case.

The data store can be any data storage device/system, but is preferably some form of a mass storage medium. For example, the data store 204 can be a magnetic storage device such as an array of Seagate disks. However, it should be noted that other types of storage media  
25 are suitable for use in the practice of the invention. For example, the data store could also be one or more remote data storage devices that are accessed over a network such as the Internet or some local area network (LAN). Another source/destination for data streaming to or from the reconfigurable logic device 202, is network 242 by way of  
30 network interface 240, as described above. In the financial industry, a network data source (e.g., the exchanges themselves, a third party provider, etc.) can provide the financial data stream 106 described above in connection with Figure 1.

The computer system defined by main processor 208 and RAM 210 is  
35 preferably any commodity computer system as would be understood by

those having ordinary skill in the art. For example, the computer system may be an Intel Xeon system or an AMD Opteron system.

The reconfigurable logic device 202 has firmware modules deployed thereon that define its functionality. The firmware socket module 220  
5 handles the data movement requirements (both command data and target data) into and out of the reconfigurable logic device, thereby providing a consistent application interface to the firmware application module (FAM) chain 230 that is also deployed on the reconfigurable logic device. The FAMs 230i of the FAM chain 230 are  
10 configured to perform specified data processing operations on any data that streams through the chain 230 from the firmware socket module 220. Preferred examples of FAMs that can be deployed on reconfigurable logic in accordance with a preferred embodiment of the present invention are described below.

15 The specific data processing operation that is performed by a FAM is controlled/parameterized by the command data that FAM receives from the firmware socket module 220. This command data can be FAM-specific, and upon receipt of the command, the FAM will arrange itself to carry out the data processing operation controlled by the received  
20 command. For example, within a FAM that is configured to compute an index value (such as the Dow Jones Industrial Average), the FAM's index computation operation can be parameterized to define which stocks will be used for the computation and to define the appropriate weighting that will be applied to the value of each stock to compute  
25 the index value. In this way, a FAM that is configured to compute an index value can be readily re-arranged to compute a different index value by simply loading new parameters for the different index value in that FAM.

Once a FAM has been arranged to perform the data processing  
30 operation specified by a received command, that FAM is ready to carry out its specified data processing operation on the data stream that it receives from the firmware socket module. Thus, a FAM can be arranged through an appropriate command to process a specified stream of data in a specified manner. Once the FAM has completed its data processing  
35 operation, another command can be sent to that FAM that will cause the

FAM to re-arrange itself to alter the nature of the data processing operation performed thereby. Not only will the FAM operate at hardware speeds (thereby providing a high throughput of target data through the FAM), but the FAMs can also be flexibly reprogrammed to  
5 change the parameters of their data processing operations.

The FAM chain 230 preferably comprises a plurality of firmware application modules (FAMs) 230a, 230b, ... that are arranged in a pipelined sequence. As used herein, "pipeline", "pipelined sequence", or "chain" refers to an arrangement of FAMs wherein the output of one  
10 FAM is connected to the input of the next FAM in the sequence. This pipelining arrangement allows each FAM to independently operate on any data it receives during a given clock cycle and then pass its output to the next downstream FAM in the sequence during another clock cycle.

A communication path 232 connects the firmware socket module 220  
15 with the input of the first one of the pipelined FAMs 230a. The input of the first FAM 230a serves as the entry point into the FAM chain 230. A communication path 234 connects the output of the final one of the pipelined FAMs 230m with the firmware socket module 220. The output of the final FAM 230m serves as the exit point from the FAM  
20 chain 230. Both communication path 232 and communication path 234 are preferably multi-bit paths.

Figure 3 depicts an exemplary framework for the deployment of applications on the system 100 of Figure 2. The top three layers of Figure 3 represent functionality that is executed in software on the  
25 computer system's general-purpose processor 208. The bottom two layers represent functionality that is executed in firmware on the reconfigurable logic device 202.

The application software layer 300 corresponds to high level functionality such as the type of functionality wherein one or more  
30 users interact with the application to define which data processing operations are to be performed by the FAMs and to define what data those data processing operations are to be performed upon.

The next layer is the module application programming interface (API) layer 302 which comprises a high level module API 302a and a low  
35 level module API 302b. The high level module API 302a can provide

generic services to application level software (for example, managing callbacks). The low level module API 302b manages the operation of the operating system (OS) level/device driver software 304. A software library interface 310 interfaces the high level module API 302a with the low level module API 302b. Additional details about this software library interface can be found in the above-referenced patent application 11/339,892.

The interface between the device driver software 304 and the firmware socket module 220 serves as the hardware/software interface 312 for the system 200. The details of this interface 312 are described in greater detail in the above-referenced patent application 11/339,892.

The interface between the firmware socket module 220 and the FAM chain 230 is the firmware module interface 314. The details of this interface are described in greater detail in the above-referenced patent application 11/339,892.

Figure 4(a) depicts a printed circuit board or card 400 that can be connected to the PCI-X bus 212 of a commodity computer system for use in a market data platform. In the example of Figure 4(a), the printed circuit board includes an FPGA 402 (such as a Xilinx Virtex II FPGA) that is in communication with a memory device 404 and a PCI-X bus connector 406. A preferred memory device 404 comprises SRAM and DRAM memory. A preferred PCI-X bus connector 406 is a standard card edge connector.

Figure 4(b) depicts an alternate configuration for a printed circuit board/card 400. In the example of Figure 4(b), a private bus 408 (such as a PCI-X bus), a network interface controller 410, and a network connector 412 are also installed on the printed circuit board 400. Any commodity network interface technology can be supported, as is understood in the art. In this configuration, the firmware socket 220 also serves as a PCI-X to PCI-X bridge to provide the processor 208 with normal access to the network(s) connected via the private PCI-X bus 408.

It is worth noting that in either the configuration of Figure 4(a) or 4(b), the firmware socket 220 can make memory 404 accessible

to the PCI-X bus, which thereby makes memory 404 available for use by the OS kernel 304 as the buffers for transfers from the disk controller and/or network interface controller to the FAMs. It is also worth noting that while a single FPGA 402 is shown on the printed circuit boards of Figures 4(a) and (b), it should be understood that multiple FPGAs can be supported by either including more than one FPGA on the printed circuit board 400 or by installing more than one printed circuit board 400 in the computer system. Figure 5 depicts an example where numerous FAMs in a single pipeline are deployed across multiple FPGAs.

As shown in Figures 2-4, inbound data (from the kernel 204 to the card 300) is moved across the bus 112 in the computer system to the firmware socket module 120 and then delivered by the firmware socket module 120 to the FAM chain 130. Outbound data (from the card 300 to the kernel 204) are delivered from the FAM chain 130 to the firmware socket module 120 and then delivered by the firmware socket module 120 across the PCI-X bus to the software application executing on the computer system. As shown in Figure 3, the three interacting interfaces that are used are the firmware module interface 214, the hardware/software interface 212, and the software library interface 210.

In an effort to improve upon conventional market data platforms, the inventors herein disclose a new market data platform architecture, an embodiment of which is shown in Figure 6. The market data platform 600 shown in Figure 6 consolidates the functional units 102 shown in Figure 1 into much fewer physical devices and also offloads much of the data processing performed by the GPPs of the functional units 102 to reconfigurable logic.

For example, with the architecture of Figure 6, the feed compressor 602 can be deployed in an appliance such as system 200 shown in Figure 2. The reconfigurable logic 202 can be implemented on a board 400 as described in connection with Figures 4(a) or 4(b). Feed compressor 602 is used to compress the content of the financial data stream 106 arriving from various individual sources. Examples of compression techniques that can be used include the open standard

"glib" as well as any proprietary compression technique that may be used by a practitioner of the present invention. Appropriate FAM modules and a corresponding FAM pipeline to implement such a feed compression operation can be carried out by a person having ordinary skill in the art using the design techniques described in connection with the above-referenced patent and patent applications and basic knowledge in the art concerning feed compression. As a result, a variety of hardware templates available for loading on reconfigurable logic can be designed and stored for use by the market data platform 600 to implement a desired feed compression operation.

Preferably, the feed compressor device 602 is deployed in a physical location as close to the feed source 106 as possible, to thereby reduce communication costs and latency. For example, it would be advantageous to deploy the feed compressor device 602 in a data center of an extranet provider (e.g., Savvis, BT Radianz, etc.) due to the data center's geographic proximity to the source of the financial market data 106. Because the compression reduces message sizes within the feed stream 106, it will be advantageous to perform the compression prior to the stream reaching wide area network (WAN) 620a; thereby improving communication latency through the network because of the smaller message sizes.

WAN 620 preferably comprises an extranet infrastructure or private communication lines for connection, on the inbound side, to the feed handlers deployed in device 604. On the outbound side, WAN 620 preferably connects with device 606, as explained below. It should be noted that WAN 620 can comprise a single network or multiple networks 620a and 620b segmented by their inbound/outbound role in relation to platform 600. It is also worth noting that a news feed with real-time news wire reports can also be fed into WAN 620a for delivery to device 604.

Device 604 can be deployed in an appliance such as system 200 shown in Figure 2. The reconfigurable logic 202 can be implemented on a board 400 as described in connection with Figures 4(a) or 4(b). Whereas the conventional GPP-based system architecture shown in Figure 1 deployed the functional units of feed handling/ticker plant, rule-

based calculation engines, an alert generation engine, options pricing, Last Value Cache (LVC) servers supplying snapshot and/or streaming interfaces, historical time-series oriented databases with analytics, and news databases with search capabilities in software on  
5 separate GPPs, the architecture of Figure 6 can consolidate these functions, either partially or in total, in firmware resident on the reconfigurable logic (such as one or more FPGAs) of device 604.

Feed handlers, which can also be referred to as feed producers, receive the real-time data stream, either compressed from the feed  
10 compressor 602 as shown in Figure 6 or uncompressed from a feed source, and converts that compressed or uncompressed stream from a source-specific format (e.g., an NYSE format) to a format that is common throughout the market data platform 600. This conversion process can be referred to as "normalization". This "normalization"  
15 can be implemented in a FAM chain that transforms the message structure, converts the based units of specific field values within each message, maps key field information to the common format of the platform, and fills in missing field information from cached or database records. In situations where the received feed stream is a  
20 compressed feed stream, the feed handler preferably also implements a feed decompression operation.

LVCs maintain a database of financial instrument records whose functionality can be implemented in a FAM pipeline. Each record represents the current state of that financial instrument in the  
25 market place. These records are updated in real-time via a stream of update messages received from the feed handlers. The LVC is configured to respond to requests from other devices for an up-to-the-instant record image for a set of financial instruments and redistribute a selective stream of update messages pertaining to those  
30 requested records, thereby providing real-time snapshots of financial instrument status. From these snapshots, information such as the "latest price" for a financial instrument can be determined, as described in the above-referenced 10/153,151 application.

Rule-based calculation engines are engines that allow a user to  
35 create his/her own synthetic records whose field values are derived

from calculations performed against information obtained from the LVC, information extracted from a stream of update messages generated from the LVC, or from alternate sources. These rule-based calculation engines are amenable to implementation in a FAM pipeline. It should  
5 also be noted that the rule-based calculation engine can be configured to create new synthetic fields that are included in existing records maintained by the LVC. The new values computed by the engine are computed by following a set of rules or formulas that have been specified for each synthetic field. For example, a rule-based  
10 calculation engine can be configured to compute a financial instrument's Volume Weighted Average Price (VWAP) via a FAM pipeline that computes the VWAP as the sum of PxS for every trade meeting criteria X, wherein P equals the trade price and wherein S equals the trade size. Criteria X can be parameterized into a FAM filter that  
15 filters trades based on size, types, market conditions, etc. Additional examples of rule-based calculations that can be performed by the rule-based calculation engine include, but are not limited to, a minimum price calculation for a financial instrument, a maximum price calculation for a financial instrument, a Top 10 list for a  
20 financial instrument or set of financial instruments, etc.

An alert generation engine can also be deployed in a FAM pipeline. Alert generation engines are similar to a rule-based calculation engine in that they monitor the current state of a financial instrument record (or set of financial instrument records),  
25 and the alert generation engine will trigger an alert when any of a set of specified conditions is met. An indication is then delivered via a variety of means to consuming applications or end users that wish to be notified upon the occurrence of the alert.

Option pricing is another function that is highly amenable to  
30 implementation via a FAM pipeline. An "option" is a derivative financial instrument that is related to an underlying financial instrument, and the option allows a person to buy or sell that underlying financial instrument at a specific price at some specific time in the future. An option pricing engine is configured to perform  
35 a number of computations related to these options and their underlying

instruments (e.g., the theoretical fair market value of an option or the implied volatility of the underlying instrument based upon the market price of the option). A wide array of computational rules can be used for pricing options, as is known in the art. Most if not all  
5 industry-accepted techniques for options pricing are extremely computation intensive which introduces significant latency when the computations are performed in software. However, by implementing option pricing in a FAM pipeline, the market data platform 600 can significantly speed up the computation of option pricing, thereby  
10 providing in important edge to traders who use the present invention.

A time series database is a database that maintains a record for each trade or quote event that occurs for a set of financial instruments. This information may be retrieved upon request and returned in an event-by-event view. Alternative views are available  
15 wherein events are "rolled up" by time intervals and summarized for each interval. Common intervals include monthly, weekly, daily, and "minute bars" where the interval is specified to be some number of minutes. The time series database also preferably compute a variety of functions against these historic views of data, including such  
20 statistical measures as volume weighted average price (VWAP), money flow, or correlations between disparate financial instruments.

A news database maintains a historical archive of news stories that have been received from a news wire feed by way of the feed handler. The news database is preferably configured to allow end  
25 users or other applications to retrieve news stories or headlines based upon a variety of query parameters. These query parameters often include news category assignments, source identifiers, or even keywords or keyword phrases. The inventors herein note that this searching functionality can also be enhanced using the search and data  
30 matching techniques described in the above-referenced patent and patent applications.

Appropriate FAM modules and corresponding FAM pipelines to implement these various functions for device 604 can be carried out by a person having ordinary skill in the art using the design techniques  
35 described in connection with the above-referenced patent and patent

applications and basic knowledge in the art concerning each function. As a result, a variety of hardware templates available for loading on reconfigurable logic can be designed and stored in memory (such as on a disk embodied by data store 204 in connection with Figure 2) for use  
5 by the market data platform 600 to implement a desired data processing function. Persistent data storage unit 630 can be accessible to device 604 as device 604 processes the feed stream in accordance with the functionality described above. Storage 630 can be embodied by data store 204 or other memory devices as desired by a practitioner of  
10 the invention.

Traders at workstations 104 (or application programs 150 running on an entity's own trading platform) can then access the streaming financial data processed by device 604 via a connection to local area network (LAN) 622. Through this LAN connection, workstations 104 (and  
15 application program 15) also have access to the data produced by devices 606, 608, 612, 614, and 616. Like devices 602 and 604, devices 606, 608, 612, 614, and 616 can also be deployed in an appliance such as system 200 shown in Figure 2, wherein the reconfigurable logic 202 of system 200 can be implemented on a board  
20 400 as described in connection with Figures 4(a) or 4(b).

Device 606 preferably consolidates the following functionality at least partially into firmware resident on reconfigurable logic: an order book server; an order router; direct market access gateways to exchanges, Electronic Communication Networks (ECNs), and other  
25 liquidity pools; trading engines; an auto-quote server; and a compliance journal.

An "order book server" is similar to a LVC in that the order book server maintains a database in memory (e.g., in memory device 404 on board 400) of financial instrument records, and keeps that database up  
30 to date in real-time via update messages received from the feed handlers. For each record, the order book server preferably maintains a sorted list of the bids and offers associated with all outstanding orders for that instrument. This list is known as the "book" for that instrument. The order information for each instrument is received  
35 from a variety of different trading venues in stream 106 and is

aggregated together to form one holistic view of the market for that particular instrument. The order book server is configured to respond to requests from workstation 104 users to present the book in a number of different ways. There are a variety of different "views",  
5 including but not limited to: a "top slice" of the book that returns orders whose prices are considered to be within a specified number of price points of the best price available in the market (the best price being considered to be the "top" of the book); a price aggregate view where orders at the same price point are aggregated together to create  
10 entries that are indicative of the total number of orders available at each price point; and an ordinary view with specific trading venues (which are the source of orders) excluded.

An order router is a function that can take a buy or sell order for a specified financial instrument, and based upon a variety of  
15 criteria associated with the order itself or the end user or application submitting the order, route the order (in whole or in part) to the most appropriate trading venue, such as an exchange, an Alternate Trading System (ATS), or an ECN.

The direct market access gateway functionality operates to relay  
20 orders to a trading venue (such as an exchange, ECN, ATS, etc.) via WAN 620b. Before sending an order out however, the gateway preferably transforms the order message to a format appropriate for the trading venue.

The trading engine functionality can also be deployed on  
25 reconfigurable logic. An algorithmic trading engine operates to apply a quantitative model to trade orders of a defined quantity to thereby automatically subdivide that trade order into smaller orders whose timing and size are guided by the goals of the quantitative model so as to reduce the impact that the original trade order may have on the  
30 current market price. Also, a black box trading engine operates to automatically generate trades by following a mathematical model that specifies relationships or conditional parameters for an instrument or set of instruments. To aid this processing, the black box trading engine is fed with real-time market data.

An auto-quote server is similar to a black box trading engine. The auto-quote server operates to automatically generate firm quotes to buy or sell a particular financial instrument at the behest of a "market maker"; wherein a "market maker" is a person or entity which  
5 quotes a buy and/or sell price in a financial instrument hoping to make a profit on the "turn" or the bid/offer spread.

A feed/compliance journal can also be implemented in a FAM pipeline. The feed/compliance journal functions to store information (in persistent storage 632) related to the current state of the entire  
10 market with regard to a particular financial instrument at the time a firm quote or trade order is submitted to a single particular marketplace. The feed/compliance journal can also provide a means for searching storage 632 to provide detailed audit information on the state of the market when a particular firm quote or trade order was  
15 submitted. The inventors herein note that this searching functionality can also be enhanced using the search and data matching techniques described in the above-referenced patent and patent applications.

As mentioned above in connection with device 604, appropriate FAM  
20 modules and corresponding FAM pipelines to implement these various functions for device 606 can be carried out by a person having ordinary skill in the art using the design techniques described in connection with the above-referenced patent and patent applications and basic knowledge in the art concerning each function. As a result,  
25 a variety of hardware templates available for loading on reconfigurable logic can be designed and stored for use by the market data platform 600 to implement a desired data processing function. Persistent data storage unit 632, which can be embodied by data store 204, can be accessible to device 606 as device 606 processes the feed  
30 stream in accordance with the functionality described above.

Device 608 preferably implements an internal matching system/engine in firmware resident on reconfigurable logic. An internal matching system/engine operates to match a buyer's bid with a seller's offer to sell for a particular financial instrument, to  
35 thereby execute a deal or trade. An indication of a completed trade

is then submitted to the appropriate reporting and settlement systems. The internal matching system/engine may create bids or offers as would a market maker in order to provide an orderly market and a minimum amount of liquidity by following a set of programmatically-defined  
5 rules.

Device 610 preferably implements an order management system (OMS) in firmware resident on reconfigurable logic. An OMS operates to facilitate the management of a group of trading accounts, typically on behalf of a broker. The OMS will monitor buy and sell orders to  
10 ensure that they are appropriate for the account owner in question based upon his/her account status, credit and risk profiles. The OMS typically incorporates a database via persistent storage 638 (which may be embodied by data store 204) used to hold account information as well as an archive of orders and other activity for each account.

Device 612 preferably implements entitlements and reporting functionality. A market data platform such as system 600 is a mechanism for distributing data content to a variety of end users. Many content providers charge on a per user basis for access to their data content. Such content providers thus prefer a market data  
15 platform to have a mechanism to prohibit (or entitle) access to specific content on an individual user basis. Entitlement systems may also supply a variety of reports that detail the usage of different content sets. To achieve this functionality, device 612, in conjunction with database 634, preferably operates to maintain a  
20 database of users, including authentication credentials and entitlement information which can be used by devices 604, 606, 608, 610 and 616 for entitlement filtering operations in conjunction with the data processing operations performed thereby.

Device 614 preferably implements management and monitoring for  
30 the market data platform 600. Management and monitoring functionality provides a means for users to operate the applications running within the platform 600 and monitor the operational state and health of individual components thereon. Preferably, the management and monitoring functionality also provides facilities for reconfiguring

the components as well as means for performing any other appropriate manual chores associated with running the platform.

Device 616 preferably implements publishing and contribution server functionality. Contribution servers (also known as publishing  
5 servers) allow users to convert information obtained from an end-user application (or some other source within his/her enterprise) into a suitable form, and to have it distributed by the market data platform 600.

As mentioned above in connection with devices 604 and 606,  
10 appropriate FAM modules and corresponding FAM pipelines to implement these various functions for devices 608, 610, 612, 614, and 616 can be carried out by a person having ordinary skill in the art using the design techniques described in connection with the above-referenced patent and patent applications and basic knowledge in the art  
15 concerning each function. As a result, a variety of hardware templates available for loading on reconfigurable logic can be designed and stored for use by the market data platform 600 to implement a desired data processing function. Persistent data storage units 634 and 636 can be accessible to devices 612 and 614  
20 respectively as those devices process the data in accordance with the functionality described above.

In deploying this functionality, at least in part, upon reconfigurable logic, the following modules/submodules of the functions described above are particularly amenable to implementation  
25 on an FPGA: fixed record format message parsing, fixed record format message generation, FIX message parsing, FIX message generation, FIX/FAST message parsing, FIX/FAST message generation, message compression, message decompression, interest and entitlement filtering, financial instrument symbol mapping, record ID mapping,  
30 price summary LVC update/retrieve/normalize (LVC), order book cache update/retrieve/normalize (OBC), generic LVC (GVC), minute bar generation, programmatic field generation (with LVC, OBC, etc.), historic record search and filter, book-based algorithmic order routing, trade order generation, and autoquote generation. It should  
35 be understood by those having ordinary skill in the art that this list

is exemplary only and not exhaustive; additional modules for financial data processing can also be employed in a FAM or FAM pipeline in the practice of the present invention.

With fixed record format message parsing, a fixed format message  
5 is decomposed into its constituent fields as defined by a programmable "data dictionary". Entries within the data dictionary describe the fields within each type of message, their positions and sizes within those messages, and other metadata about the field (such as data type, field identifiers, etc.). Preferably, the data dictionary is stored  
10 in persistent storage such as data store 204 of the system 200. Upon initialization of the FAM pipeline on board 400, the data dictionary is then preferably loaded into memory 404 for usage by the FAM pipeline during data processing operations.

With fixed record format message generation, a fixed format  
15 message is generated by concatenating the appropriate data representing fields into a message record. The message structure and format is described by a programmable data dictionary as described above.

With FIX message parsing, a FIX-formatted message is decomposed  
20 into its constituent fields as defined by a programmable data dictionary as described above; FIX being a well-known industry standard for encoding financial message transactions.

With FIX message generation, a FIX-formatted message is generated  
by concatenating the appropriate data representing the fields into a  
25 FIX message record. Once again, the message structure and format is described by a programmable data dictionary as described above.

With FIX/FAST message parsing, a FIX and/or FAST message (FAST  
being a well known variation of FIX) is decomposed into its  
constituent fields as defined by a programmable data dictionary as  
30 described above.

With FIX/FAST message generation, a FIX-formatted and/or FAST-  
formatted message is generated by concatenating the appropriate data  
representing fields into a FIX/FAST message record. The message  
structure and format is defined by a programmable data dictionary as  
35 described above.

With message compression, a message record is compressed so as to require less space when contained in a memory device and to require less communication bandwidth when delivered to other systems. The compression technique employed is preferably sufficient to allow for reconstruction of the original message when the compressed message is processed by a corresponding message decompression module.

With interest and entitlement filtering, a stream of messages coming from a module such as one of the caching modules described below (e.g., price summary LVC, order book OBC, or generic GVC) is filtered based upon a set of entitlement data and interest data that is stored for each record in the cache. This entitlement and interest data defines a set of users (or applications) that are both entitled to receive the messages associated with the record and have expressed an interest in receiving them. This data can be loaded into memory from storage 634 during initialization of the board 400.

With financial instrument symbol mapping, a common identifying string for a financial instrument (typically referred to as the "symbol") is mapped into a direct record key number that can be used by modules such as caching modules (LVC, OBC, GVC) to directly address the cache record associated with that financial instrument. The record key number may also be used by software to directly address a separate record corresponding to that instrument that is kept in a storage, preferably separate from board 400.

With record ID mapping, a generic identifying string for a record is mapped into a direct record key number that can be used by a caching module (e.g., LVC, OBC, GVC) or software to directly address the record in a storage medium.

The price summary Last Value Cache update/retrieve/normalize (LVC) operation operates to maintain a cache of financial instrument records whose fields are updated in real-time with information contained in streaming messages received from a message parsing module. The type of update performed for an individual field in a record will be defined by a programmable data dictionary as described above, and may consist of moving the data field from the message to the record, updating the record field by accumulating the data field

over a series of messages defined within a time-bounded window,  
updating the record field only if certain conditions as defined by a  
set of programmable rules are true, or computing a new value based  
upon message and/or record field values as guided by a programmable  
5 formula.

The order book cache update, retrieve and normalize (OBC)  
operation operates to maintain a cache of financial instrument records  
where each record consists of an array of sub-records that define  
individual price or order entries for that financial instrument. A  
10 sort order is maintained for the sub-records by the price associated  
with each sub-record. The fields of the sub-records are updated in  
real-time with information contained in streaming messages received  
from a message parsing module. Sub-records associated with a record  
are created and removed in real-time according to information  
15 extracted from the message stream, and the sort order of sub-records  
associated with a given record is continuously maintained in real-  
time. The type of update performed for an individual field in a sub-  
record will be defined by a programmable data dictionary, and may  
consist of moving the data field from the message to the sub-record,  
20 updating the sub-record field by accumulating the data field over a  
series of messages defined within a time-bounded window, updating the  
sub-record field only if certain conditions as defined by a set of  
programmable rules are true, or computing a new value based upon  
message and/or record or sub-record fields as guided by a programmable  
25 formula.

The Generic Last Value Cache (GVC) operation operates to maintain  
a cache of records whose fields are updated in real-time with  
information contained in streaming messages received from a message  
parsing module. The structure of a record and the fields contained  
30 within it are defined by a programmable data dictionary, as described  
above. The type of update performed for an individual field in a  
record will be defined by a programmable data dictionary, and may  
consist of moving the data field from the message to the record,  
updating the record field by accumulating the data field over a series  
35 of messages defined within a time-bounded window, updating the record

field only if certain conditions as defined by a set of programmable rules are true, or computing a new value based upon message and/or record field values as guided by a programmable formula.

5 A minute bar generation operation operates to monitor real-time messages from a message parsing module or last value cache module for trade events containing trade price information and create "minute bar" events that summarize the range of trade prices that have occurred over the previous time interval. The time interval is a programmable parameter, as is the list of records for which minute  
10 bars should be generated, and the fields to include in the generated events.

A Top 10 list generation operation operates to monitor real-time messages from a message parsing module or last value cache module for trade events containing price information and create lists of  
15 instruments that indicate overall activity in the market. Such lists may include (where 'N' is programmatically defined): top N stocks with the highest traded volume on the day; top N stocks with the greatest positive price change on the day; top N stocks with the largest percentage price change on the day; top N stocks with the greatest  
20 negative price change on the day; top N stocks with the greatest number of trade events recorded on the day; top N stocks with the greatest number of "large block" trades on the day, where the threshold that indicates whether a trade is a large block trade is defined programmatically.

25 A programmatic field generation (via LVC, OBV, GVC, etc.) operation operates to augment messages received from a message parsing module with additional fields whose values are defined by a mathematical formula that is supplied programmatically. The formula may reference any field within the stream of messages received from a  
30 message parsing module, any field contained within a scratchpad memory associated with this module, or any field contained within any record held within any the record caches described herein.

A programmatic record generation (with LVC, OBC, GVC, etc.) operation operates to generate records that represent synthetic  
35 financial instruments or other arbitrary entities, and a series of

event messages that signal a change in state of each record when the record is updated. The structure of the records and the event messages are programmatically defined by a data dictionary. The field values contained with the record and the event messages are defined by  
5 mathematical formulas that are supplied programmatically. The formulas may reference any field within the stream of messages received from a message parsing module, any field contained within a scratchpad memory associated with this module, or any field contained within any record held within any the record caches described herein.  
10 Updates to field values may be generated upon receipt of a message received from another module, or on a time interval basis where the interval is defined programmatically.

A historic record search and filter operation operates to filter messages received from a message parsing module that represent a time  
15 series of events to partition the events into various sets, where each set is defined as a collection of criteria applied to event attributes. The event message structure, criteria and attributes are all programmatically defined. Event attributes include, but are not limited to: financial instrument symbol, class of symbol, time and  
20 date of event, type of event, or various indicator fields contained within the event. Multiple events within a set may be aggregated into a single event record according to a collection of aggregation rules that are programmatically defined and applied to attributes of the individual events. Aggregation rules may include, but are not limited  
25 to, aggregating hourly events into a single daily event, aggregating daily events into a single weekly event, or aggregating multiple financial instruments into a single composite instrument.

These functions (as well as other suitable financial data processing operations) as embodied in FAMs can then be combined to  
30 form FAM pipelines that are configured to produce useful data for a market data platform. For example, a feed compressor FAM pipeline can employ FAMs configured with the following functions: fixed record format message parsing, fixed record format message generation, FIX message parsing, FIX message generation, FIX/FAST message parsing,

FIX/FAST message generation, message compression, and message decompression.

Figure 7 illustrates an exemplary FAM pipeline for performing a FIX-to-FAST message transformation. FAM 702 is configured to receive an incoming stream of FIX messages and perform FIX message parsing thereon, as described above. Then the parsed message is passed to FAM 704, which is configured to field decode and validate the parsed message. To aid this process, FAM 704 preferably has access to stored templates and field maps in memory 710 (embodied by memory 404 on board 400). The templates identify what fields exist in a given message type, while the field maps uniquely identify specific fields in those message (by location or otherwise). Next, FAM 704 provides its output to FAM 706, which is configured to perform a FAST field encode on the received message components. Memory 710 and any operator values stored in memory 712 aid this process (memory 712 also being embodied by memory 404 on board 400). The operator values in memory 712 contains various state values that are preserved from one message to the next, as defined by the FAST encoding standard. Then, the encoded FAST messages are serialized by FAM 708 to form a FAST message stream and thereby complete the FIX to FAST transformation.

Figure 8 illustrates an exemplary FAM pipeline for performing a FAST-to-FIX message transformation. An incoming FAST stream is received by FAM 802, which deserializes the stream of FAST messages. The deserialized FAST messages are then provided to FAM 804, which operates to decode the various fields of the FAST messages as aided by the templates and field maps in memory 812 and the operator values in memory 810. Thereafter, FAM 806 is preferably configured to perform message query filtering. Message query filters allow for certain messages to be excluded from the message flow. Such filters are preferably parameterized in FAM 806 such that filtering criteria based on the field values contained within each message can be flexibly defined and loaded onto the FPGA. Examples of filtering criteria that can be used to filter messages include a particular type of instrument (e.g., common stock, warrant, bond, option, commodity, future, etc.), membership within a prescribed set of financial instruments (e.g., an

index or "exchange traded fund" (ETF)), message type, etc. Next, FAM 808 operates to perform FIX message generation by appropriately encoding the various message fields, as aided by the templates and field maps in memory 812. Thus, the FAM pipeline shown in Figure 8 operates to transform FAST message to FIX messages. Memory units 810 and 812 are preferably embodied by memory 404 of board 400.

Figure 9 depicts an exemplary FAM pipeline for carrying out a variety of data processing tasks. The FAM pipeline of Figure 9 takes in a FAST message stream. FAMS 902 and 904 operate in the same manner as FAMS 802 and 804 in Figure 8. Thus, the output of FAM 904 comprises the data content of the FAST message decomposed into its constituent fields. This content is then passed to a variety of parallel FAMS 906, 908, and 910. FAM 906 performs a administrative record filter on the data it receives. The administrative record filter preferably operates to pass through message types that are not processed by any of the other FAM modules of the pipeline. FAM 908 serves as a Top 10 lists engine, as described above. FAM 910 serves as a message query filter, as described above.

The output of FAM 910 is then passed to FAM 912, which is configured as a rule-based calculation engine, as described above. FAM 912 also receives data from a real time field value cache 926 to obtain LVC data, as does the top 10 list FAM 908. Cache 926 is preferably embodied by storage 632. The output from the rule-based calculation engine FAM 912 is then passed to parallel FAMS 914, 916, and 918. FAM 914 serves as a message multiplexer, and receives messages from the outputs of FAMS 906, 908 and 912. FAM 920 receives the messages multiplexed by FAM 914, and serves to encode those messages to a desired format. FAM 916 serves as an alert engine, whose function is explained above, and whose output exits the pipeline. FAM 918 serves as a value cache update engine to ensuring that cache 926 stays current.

Figure 10 depicts another exemplary FAM pipeline for carrying out multiple data processing tasks. FAM 1002 takes in a stream of fixed format messages and parses those messages into their constituent data fields. The output of FAM 1002 can be provided to FAM 1004 and FAM

1018. FAM 1018 serves as a message synchronization buffer. Thus, as the fields of the original parsed message are passed directly from FAM 1002 to FAM 1018, FAM 1018 will buffer those data fields while the upper path of Figure 10 (defined by FAMs 1004, 1006, 1008, 1010, 1012, and 1014) process select fields of the parsed message. Thus, upon completion of the processing performed by the FAMs of the upper path, the message formatting FAM 1016, can generate a new message for output from the pipeline using the fields as processed by the upper path for that parsed message as well as the fields buffered in FAM 1018. The message formatter 1016 can then append the fields processed by the upper path FAMs to the fields buffered in FAM 1018 for that message, replace select fields buffered in FAM 1018 for that message with fields processed by the upper path FAMs, or some combination of this appending and replacing.

15 FAM 1004 operates to map the known symbol for a financial instrument (or set of financial instruments) as defined in the parsed message to a symbology that is internal to the platform (e.g., mapping the symbol for IBM stock to an internal symbol "12345"). FAM 1006 receives the output from FAM 1004 and serves to update the LVC cache via memory 1024. The output of FAM 1006 is then provided in parallel to FAMs 1008, 1010, 1012, and 1014.

FAM 1008 operates as a Top 10 list generator, as described above. FAM 1010 operates as a Minute Bar generator, as described above. FAM 1012 operates as an interest/entitlement filter, as described above, and FAM 1014 operates as a programmatic calculation engine, as described above. The outputs from FAMs 1008, 1010, 1012 and 1014 are then provided to a message formatter FAM 1016, which operates as described above to construct a fixed format message of a desired format from the outputs of FAMs 1008, 1010, 1012, 1014 and 1018.

30 In performing these tasks, FAM 1004 is aided by memory 1020 that stores templates and field maps, as well as memory 1022 that stores a symbol index. FAM 1006 is also aided by memory 1020 as well as memory 1024 which serves as an LVC cache. Memory 1020 is also accessed by FAM 1008, while memory 1024 is also accessed by FAM 1014. FAM 1012

accesses interest entitlement memory 1026, as loaded from storage 634 during initialization of the board 400.

While these figures illustrate several embodiments of FAM pipelines that can be implemented to process real time financial data streams, it should be noted that numerous other FAM pipelines could be readily devised and developed by persons having ordinary skill in the art following the teachings herein.

Further still it should be noted that for redundancy purposes and/or scaling purposes, redundant appliances 604, 606, 608, 610, 612, 614 and 616 can be deployed in a given market data platform 600.

Furthermore, it should also be noted that a practitioner of the present invention may choose to deploy less than all of the functionality described herein in reconfigurable logic. For example, device 604 may be arranged to perform only options pricing in reconfigurable logic, or some other subset of the functions listed in Figure 6. If a user later wanted to add additional functionality to device 604, it can do so by simply re-configuring the reconfigurable logic of system 200 to add any desired new functionality. Also, the dashed boxes shown in Figure 6 enclose data processing functionality that can be considered to belong to the same category of data processing operations. That is, devices 612 and 614 can be categorized as management operations. Device 604 can be categorized as providing feed handling/processing for data access, value-added services, and historic services. Devices 606, 608 and 610 can be categorized as direct market access trading systems. As improvements to reconfigurable logic continues over time such that more resources become available thereon (e.g., more available memory on FPGAs), the inventors envision that further consolidation of financial data processing functionality can be achieved by combining data processing operations of like categories, as indicated by the dashed boxes, thereby further reducing the number of appliances 200 needed to implement platform 600. Further still, in the event of such resource improvements over time for FPGAs, it can be foreseen that even further consolidation occur, including consolidation of all functionality shown in Figure 6 on a single system 200.

Thus, a platform 600 developed in the practice of the invention can be designed to improve data processing speeds for financial market information, all while reducing the number of appliances needed for platform 600 (relative to conventional GPP-based systems) as well as  
5 the space consumed by such a platform. With a platform 600, a user such as a trader at a work station 104 (or even a customer-supplied application software program 150 that accesses the platform via an application programming interface (API), can obtain a variety of information on the financial markets with less latency than would be  
10 expected from a conventional system. This improvement in latency can translate into tremendous value for practitioners of the invention.

Sample claims of various inventive aspects of the disclosed invention, not to be considered as exhaustive or limiting, all of which are fully described so as to satisfy the written description,  
15 enablement, and best mode requirement of the Patent Laws, are as follows:

1. A method for processing financial market data, the method comprising:  
20 streaming a financial market data feed through a reconfigurable logic device; and  
processing the streaming financial market data with firmware resident on the reconfigurable logic device.
- 25 2. The method of claim 1 wherein the financial market data feed comprises a financial market source data feed, the method further comprising:  
receiving the financial market source data feed from a data source.  
30
3. The method of claim 1 wherein the financial market data feed comprises a financial market secondary data feed.
4. The method of claim 1 wherein the processing step comprises:

processing the streaming financial market data through a firmware application module pipeline that is resident on the reconfigurable logic device.

- 5     5.     The method of claim 4 wherein the financial market data feed comprises a plurality of messages, and wherein the processing step further comprises:  
             normalizing the messages to a common format.
- 10    6.     The method of claim 4 wherein the processing step further comprises:  
             converting the financial market data to a plurality of messages having a predetermined format.
- 15    7.     The method of claim 4 wherein the financial market data feed comprises a plurality of messages, and wherein the processing step further comprises:  
             parsing the messages into a plurality of data fields.
- 20    8.     The method of claim 7 wherein the financial market data feed comprises a plurality of messages, and wherein the processing step further comprises:  
             performing a calculation operation on the data fields of the  
25                parsed messages.
9.     The method of claim 4 wherein the financial market data feed comprises a plurality of messages, and wherein the processing step further comprises:  
             transforming the messages from a first format to a second format.
- 30    10.    The method of claim 9 wherein the first format comprises FIX and wherein the second format comprises FAST.
11.    The method of claim 9 wherein the first format comprises FAST and  
35                wherein the second format comprises FIX.

12. The method of claim 9 wherein the first format comprises a format as received from an external feed source, and wherein the second format comprises FAST.

5

13. The method of claim 4 wherein the firmware application module pipeline is configured to perform at least one data processing operation selected from the group consisting of: a feed compression operation, a feed decompression operation, a feed handler operation, a rule-based calculation operation, a feed journal operation, an alert generation engine, an options pricing operation, a Last Value Cache (LVC) server operation, a historical time-series oriented databases with analytics operation, a news database operation with a search capability, an order book server operation, an order router operation, a direct market access gateway operation, a trading engine, an auto-quote server operation, a compliance journal operation, an internal matching system operation, an order management system operation, an entitlements and reporting operation, a management and monitoring operation, and a publishing and contribution server operation.

20

14. The method of claim 4 further comprising:  
providing data processed through the reconfigurable logic device to a trader workstation user interface.

25

15. A device for processing financial market data, the device comprising:  
a reconfigurable logic device configured to receive a data stream comprising a financial market data feed and process the streaming financial market data feed with firmware.

30

16. The device of claim 15 wherein the financial market data feed comprises a financial market source data feed, and wherein the reconfigurable logic device is further configured to receive the financial market source data feed from a data source.

35

17. The device of claim 15 wherein the financial market data feed comprises a financial market secondary data feed.
18. The device of claim 15 wherein the reconfigurable logic device is  
5 further configured to process the data stream through a firmware application module pipeline that is resident thereon.
19. The device of claim 18 wherein the financial market data feed comprises a plurality of messages, and wherein the reconfigurable  
10 logic device is further configured to normalize the messages to a common format.
20. The device of claim 18 wherein the reconfigurable logic device is further configured to convert the financial market data feed to a  
15 plurality of messages having a predetermined format.
21. The device of claim 18 wherein the financial market data feed comprises a plurality of messages, and wherein the reconfigurable logic device is further configured to parse the messages into a  
20 plurality of data fields.
22. The device of claim 21 wherein the financial market data feed comprises a plurality of messages, and wherein the reconfigurable logic device is further configured to perform a calculation operation  
25 on the data fields of the parsed messages.
23. The device of claim 18 wherein the financial market data feed comprises a plurality of messages, and wherein the reconfigurable logic device is further configured to transform the messages from a  
30 first format to a second format.
24. The device of claim 23 wherein the first format comprises FIX and wherein the second format comprises FAST.

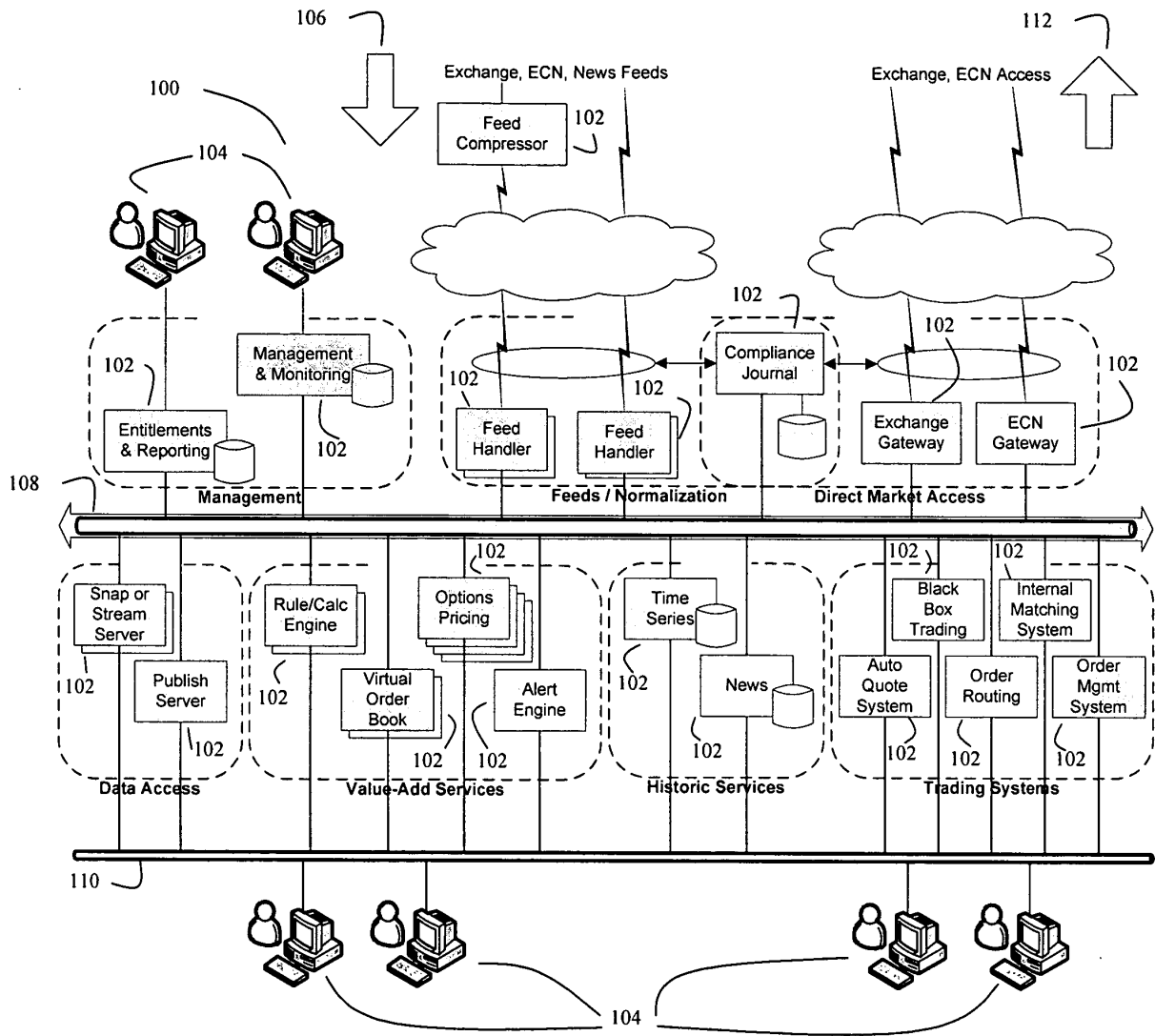
25. The device of claim 23 wherein the first format comprises FAST and wherein the second format comprises FIX.

26. The device of claim 18 wherein the firmware application module  
5 pipeline is configured to perform at least one data processing  
operation selected from the group consisting of: a feed compression  
operation, a feed decompression operation, a feed handler operation, a  
rule-based calculation operation, a feed journal operation, an alert  
generation engine, an options pricing operation, a Last Value Cache  
10 (LVC) server operation, a historical time-series oriented databases  
with analytics operation, a news database operation with a search  
capability, an order book server operation, an order router operation,  
a direct market access gateway operation, a trading engine, an auto-  
quote server operation, a compliance journal operation, an internal  
15 matching system operation, an order management system operation, an  
entitlements and reporting operation, a management and monitoring  
operation, and a publishing and contribution server operation.

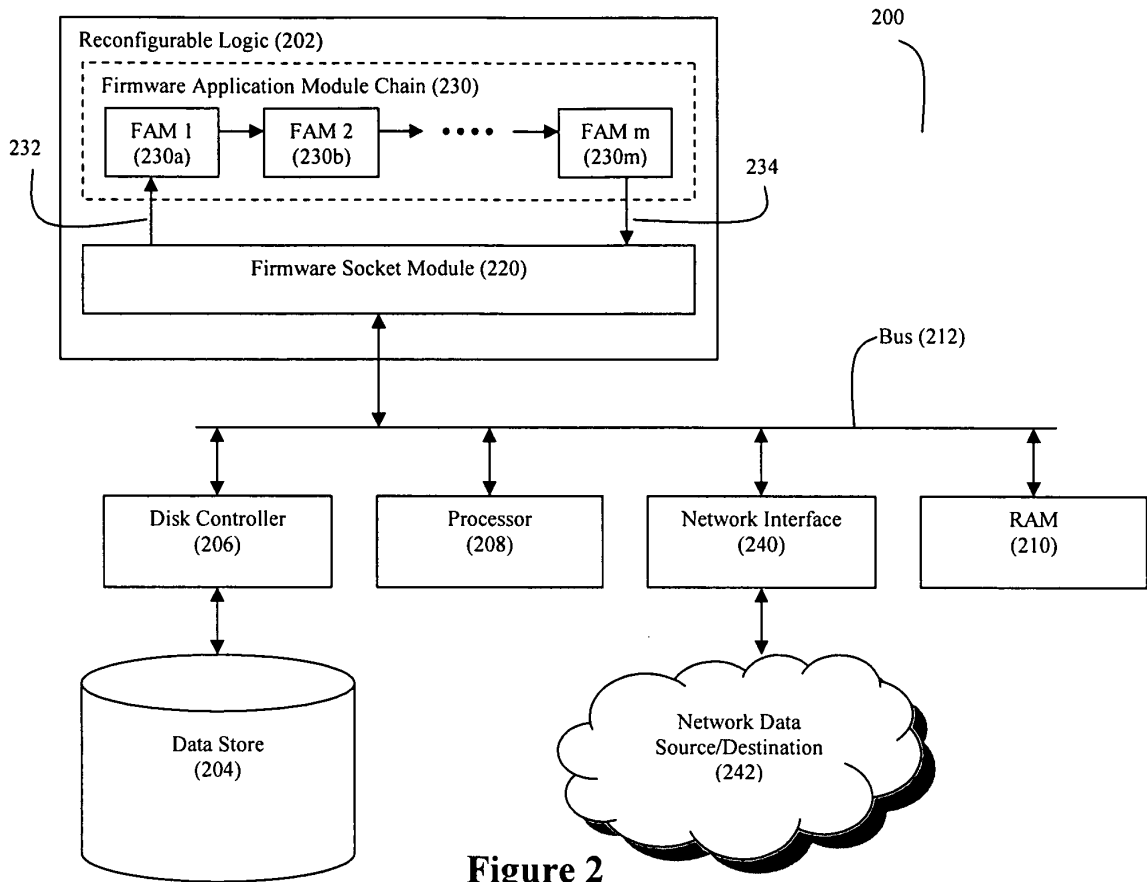
27. The device of claim 26 further comprising a plurality of data  
20 processing boards in communication with each other, each board  
comprising at least one reconfigurable logic device as defined in  
claim 21.

28. The device of claim 18 further comprising:  
25 a trader workstation in communication with the reconfigurable  
logic device, and wherein a user interface on the workstation is  
configured to display data processed by the reconfigurable logic  
device.

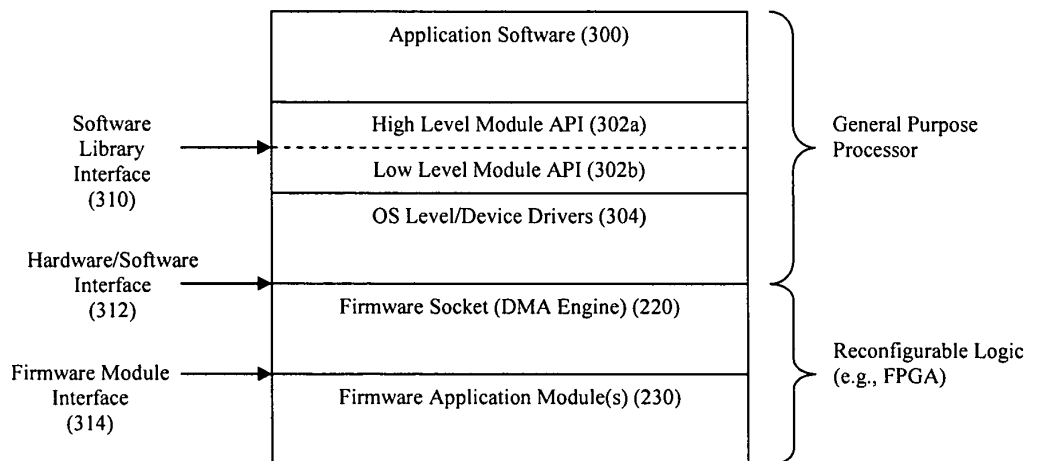
29. The device of claim 18 further comprising:  
30 an application programming interface (API) and customer-supplied  
application software in communication with the reconfigurable logic  
device, and wherein the customer-supplied application software is  
configured to receive data from the reconfigurable logic device via  
35 the API and further process the received data.



**Figure 1**  
**PRIOR ART**



**Figure 2**



**Figure 3**

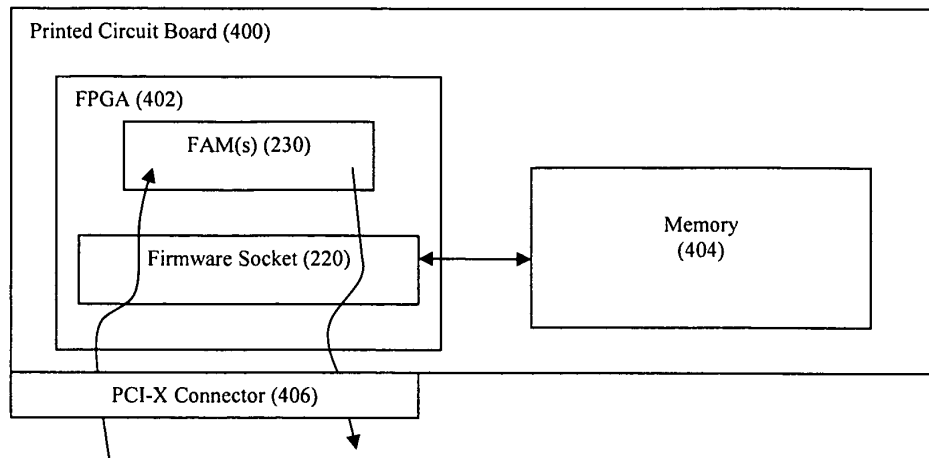


Figure 4(a)

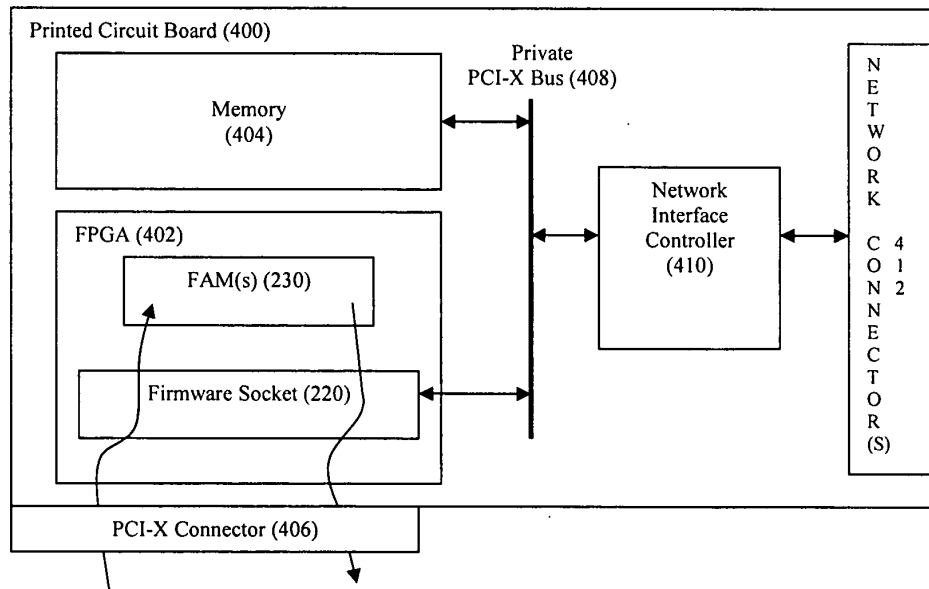
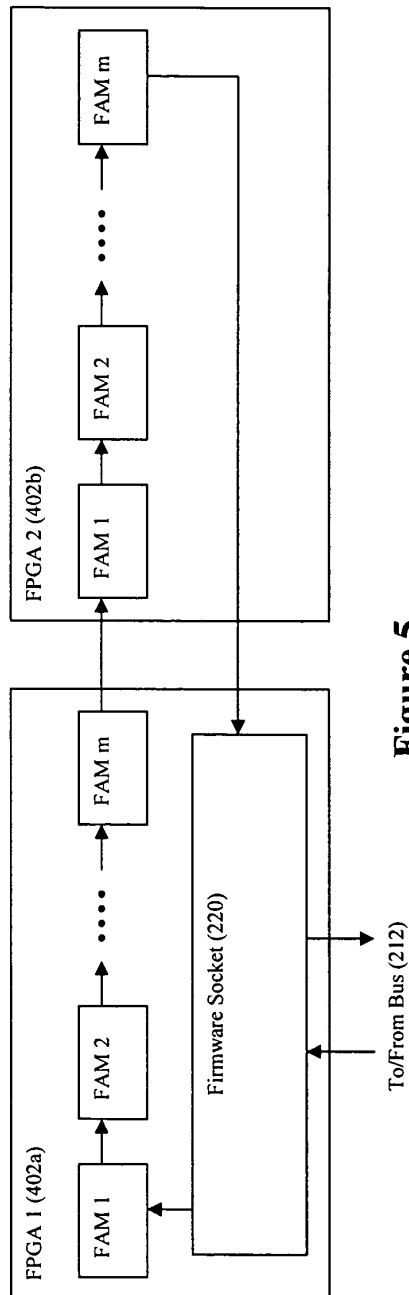


Figure 4(b)



**Figure 5**

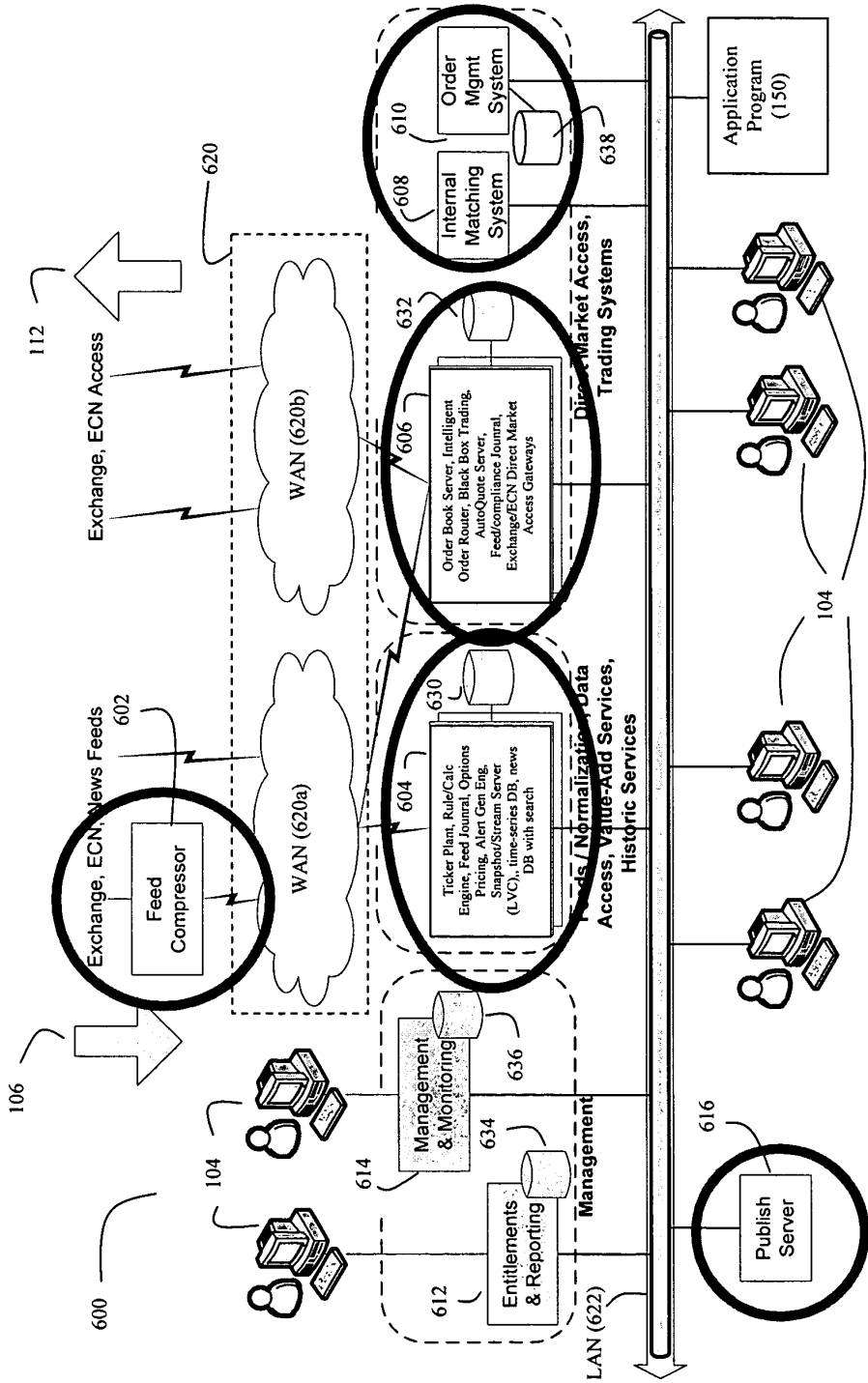


Figure 6

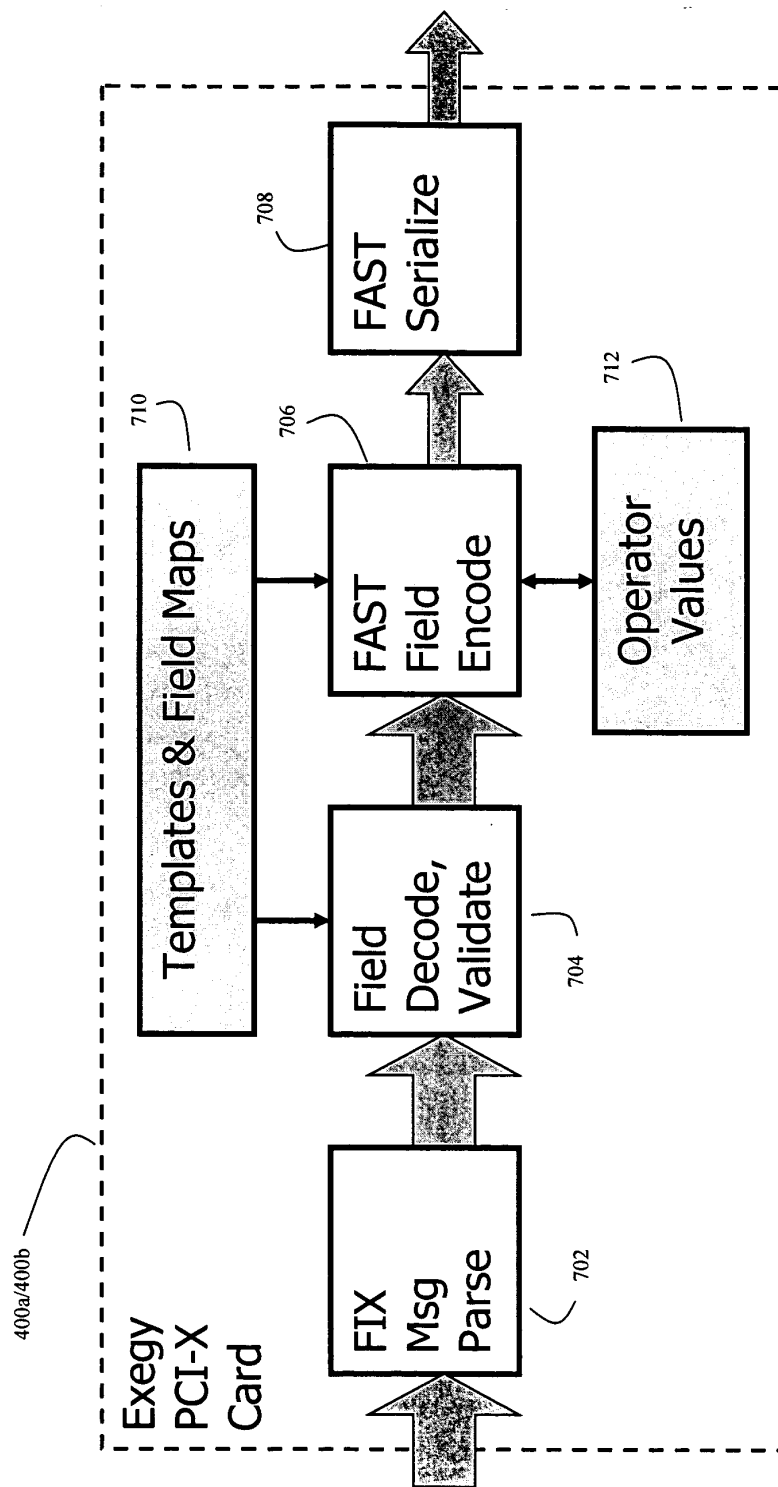
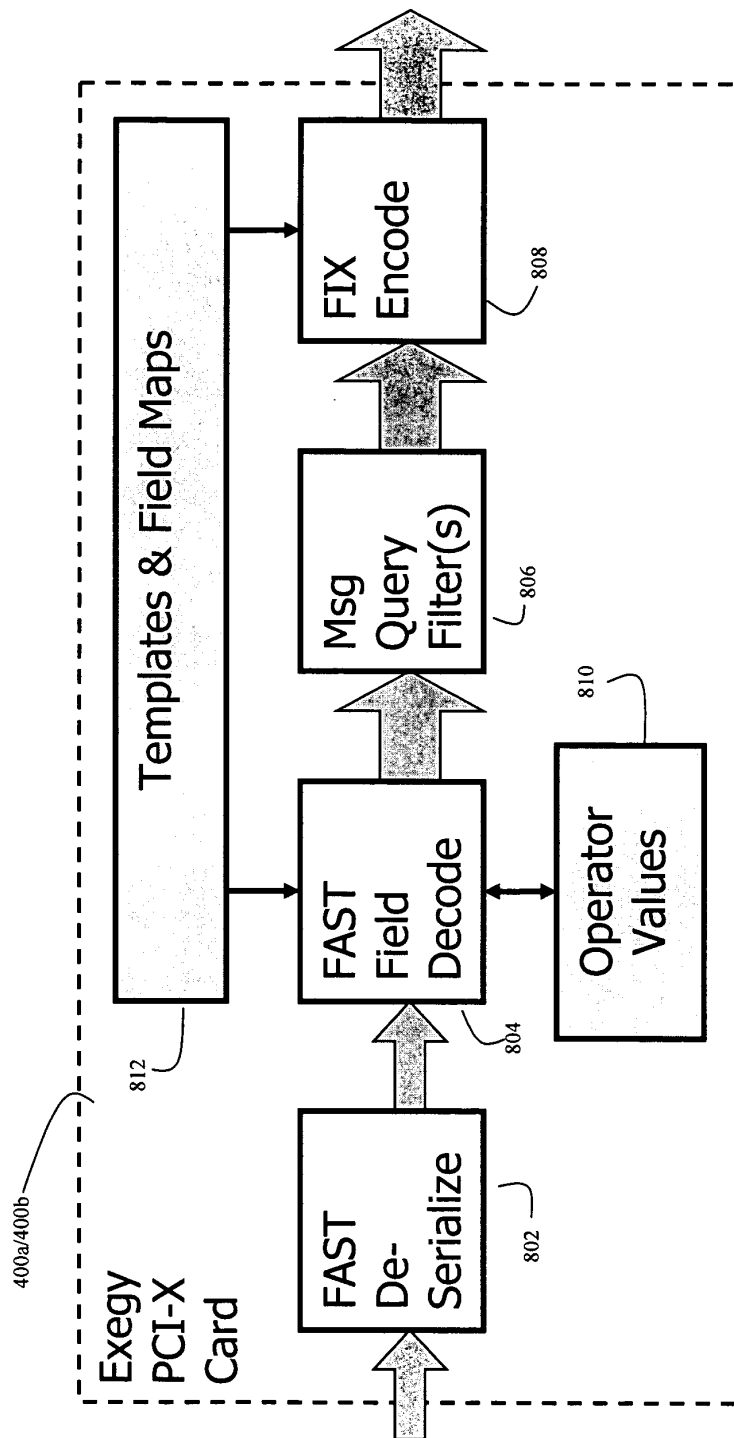


Figure 7



**Figure 8**

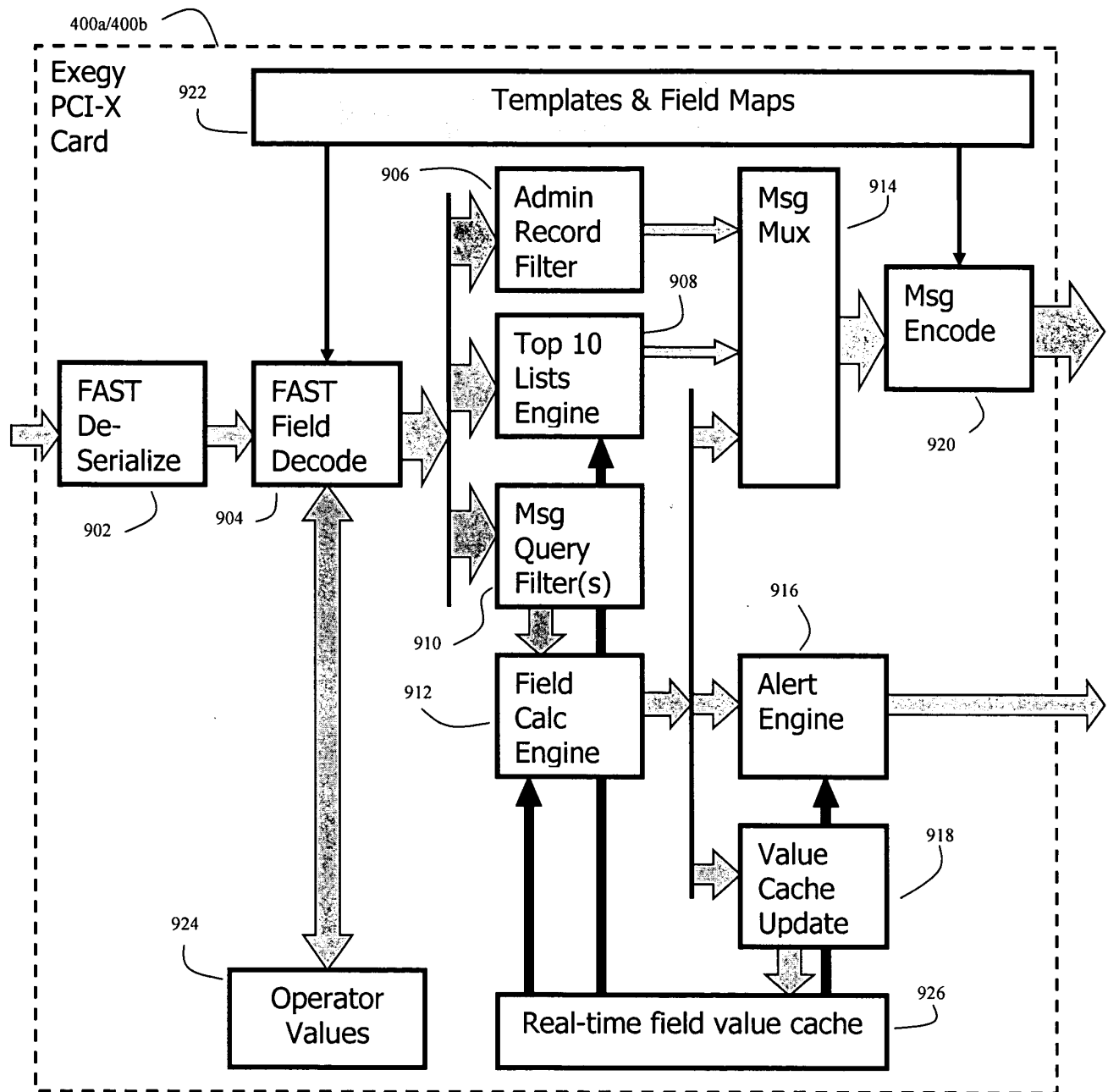


Figure 9

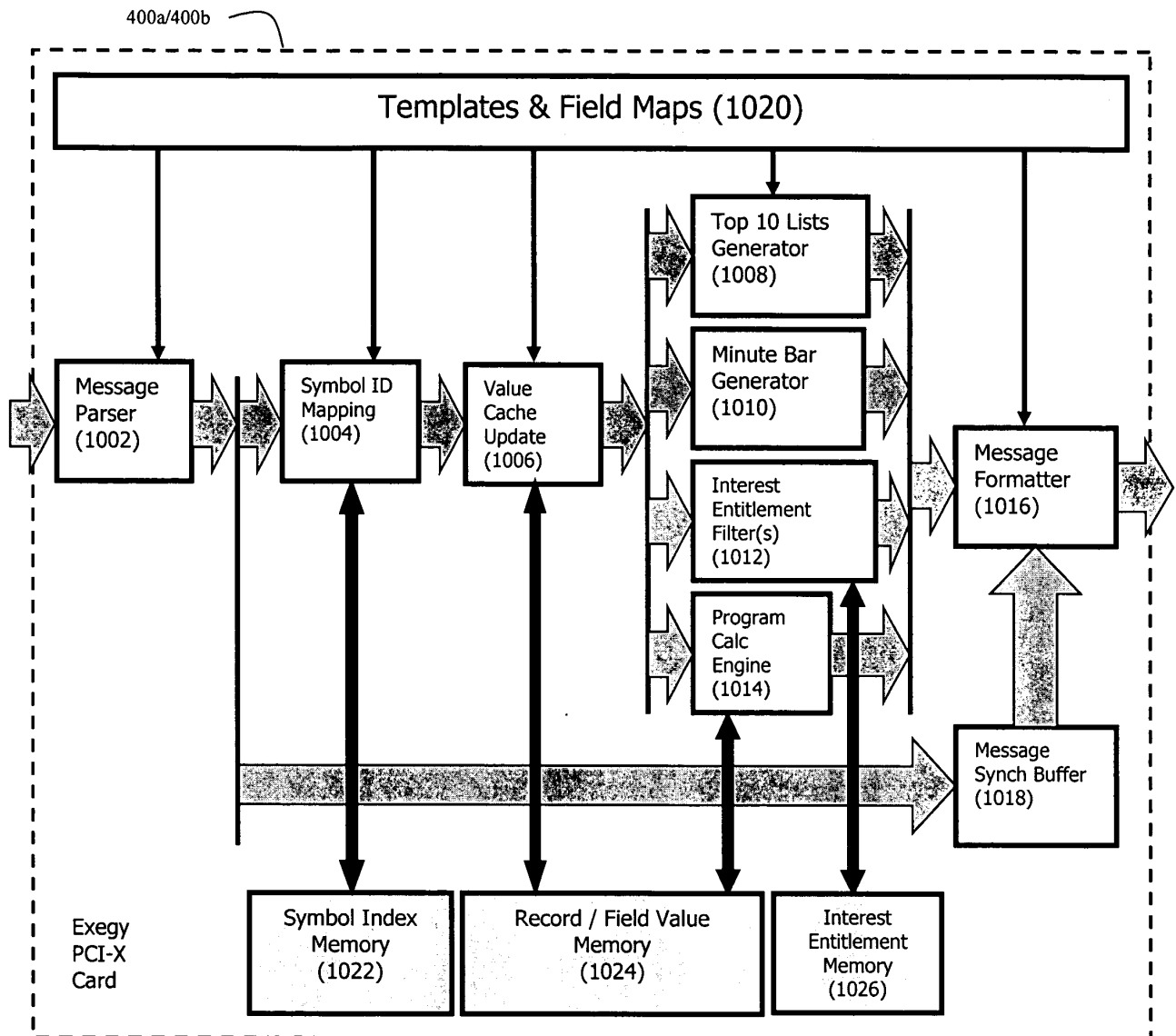


Figure 10

PATENT APPLICATION SERIAL NO. \_\_\_\_\_

U.S. DEPARTMENT OF COMMERCE  
PATENT AND TRADEMARK OFFICE  
FEE RECORD SHEET

06/21/2006 CNEGA1 00000041 200823 60614756  
01 FC:2005 100.00 DA

PTO-1556  
(5/87)

U.S. Government Printing Office: 2002 — 489-267/69033

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**POWER OF ATTORNEY  
and  
CORRESPONDENCE ADDRESS  
INDICATION FORM**

Application Number	60/814,796
Filing Date	June 19, 2006
First Named Inventor	Chamberlain et al.
Title	High Speed Processing of Financial Information Using FPGA Devices
Art Unit	
Examiner Name	
Attorney Docket Number	44826/62269

I hereby appoint:

☒ Practitioners at Customer Number

021888

OR

☐ Practitioner(s) named below:

Name	Registration Number

as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Please recognize or change the correspondence address for the above-identified application to:

☐ The address associated with the above-mentioned Customer Number:

OR

☒ The address associates with Customer Number:

021888

OR


<input type="checkbox"/> Firm or Individual Name	Benjamin L. Volk, Jr.				
Address	Thompson Coburn LLP				
Address	One US Bank Plaza				
City	St. Louis	State	MO	Zip	63101
Country	USA				
Telephone	314-552-6000	Fax	314-552-7000		

I am the:

☐ Applicant/Inventor.☒ Assignee of record of the entire interest. See 37 CFR 3.71.

Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).

**SIGNATURE of Applicant or Assignee of Record**

Name	EXAGG, Inc. by J. J. Stupp, CFO				
Signature					
Date	10/24/06	Telephone	314 450 5352		

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below\*.

☐ \*Total of \_\_\_\_\_ forms are submitted.

This collection of information is required by 37 CFR 1.131 and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**STATEMENT UNDER 37 CFR 3.73(b)**Applicant/Patent Owner: Chamberlain et al.Application No./Patent No.: 60/814,796Filed/Issue Date: June 19, 2006Entitled: High Speed Processing of Financial Information Using FPGA DevicesExegy Incorporated

(Name of Assignee)

, a corporation of the State of Delaware

(Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that it is:

1. ☒ the assignee of the entire right, title, and interest; or
2. ☐ an assignee of less than the entire right, title and interest.  
The extent (by percentage) of its ownership interest is 100 %  
in the patent application/patent identified above by virtue of either:

A. [x] An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel \_\_\_\_\_, Frame \_\_\_\_\_, or for which a copy thereof is attached.

OR

B. [ ] A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as shown below:

1. From \_\_\_\_\_ To: \_\_\_\_\_  
The document was recorded in the United States Patent and Trademark Office at  
Reel \_\_\_\_\_, Frame \_\_\_\_\_, or for which a copy thereof is attached.
2. From \_\_\_\_\_ To: \_\_\_\_\_  
The document was recorded in the United States Patent and Trademark Office at  
Reel \_\_\_\_\_, Frame \_\_\_\_\_, or for which a copy thereof is attached.
3. From \_\_\_\_\_ To: \_\_\_\_\_  
The document was recorded in the United States Patent and Trademark Office at  
Reel \_\_\_\_\_, Frame \_\_\_\_\_, or for which a copy thereof is attached.

[ ] Additional documents in the chain of title are listed on a supplemental sheet.

- [ ] Copies of assignments or other documents in the chain of title are attached.  
[NOTE: A separate copy (i.e., the original assignment document or a true copy of the original document) must be submitted to Assignment Division in accordance with 37 CFR Part 3, if the assignment is to be recorded in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

6/27/06  
Date314 450 5352  
Telephone numberJ.J. Stapp  
Typed or printed name  
SignatureCFO  
Title

This collection of information is required by 37 CFR 3.73(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

**PATENT**

ASSIGNMENT

WHEREAS, we, Roger D. Chamberlain of 64 Notre Dame Drive, St. Louis, Missouri 63141; Scott Parsons of 10 Beaufort Court, St. Charles, Missouri 63301; David E. Taylor of 3448 Missouri Avenue, St. Louis, Missouri 63118; Rod Arbaugh of 232 N. Kingshighway #1203, St. Louis, Missouri 63108; and Mark A. Franklin of 7456 Stratford Avenue, St. Louis, Missouri 63130, have invented certain new and useful improvements for the *High Speed Processing of Financial Information Using FPGA Devices* for which we filed a provisional patent application in the United States identifying us as the only inventors thereof said application being assigned serial number 66/814796 by the U.S. Patent and Trademark Office and identified as Thompson Coburn LLP, Docket 44826/62269;

WHEREAS, Exegy Incorporated, a corporation of the State of Delaware having its principal office and place of business at 3668 South Geyer Road, Suite 300, St. Louis, Missouri 63127, is desirous of acquiring the entire right, title and interest in and to said invention and improvements as disclosed in said application, in and to said application, and in and to the patent to be obtained thereon; and

NOW THEREFORE, in consideration of the sum of One Dollar (\$1.00) to us in hand paid, and other good and valuable consideration, the receipt and adequacy of all of which are hereby acknowledged, we, Roger D. Chamberlain, Scott Parsons, David E. Taylor, Rod Arbaugh, and Mark A. Franklin, have sold, assigned and transferred and do hereby sell, assign and transfer unto said Exegy Incorporated, its successors or assigns, the entire right, title and interest in and to said invention and improvements as the same are disclosed in said application which was executed by us in and to said application, and in and to any Letters Patent, including divisionals, continuations, reissues, reexaminations and extensions to be obtained therefor in the United States and all foreign countries; and we further agree to cooperate with the assignee hereunder in the obtaining and sustaining of any and all such Letters Patent as the assignee may desire to apply for, but at the expense of said assignee.

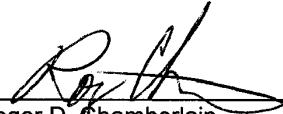
The Commissioner of Patents or other proper official is hereby authorized and requested to issue said Letters Patent to said Exegy Incorporated in accordance with this Assignment.

The undersigned hereby authorize and request the attorneys of record in said application(s) to insert in this Assignment the serial number, filing date, and date of execution of said application(s).

For said considerations, the undersigned hereby agree, upon the request and at the expense of said Assignees, to execute any divisional, continuation, continuation-in-part or substitute application for said invention or improvements, and any oath or affidavit relating thereto, and any application for the reissue or extension of any Letters Patent that may be granted upon said application(s), and, in the event of any application or Letters Patent assigned herein becoming involved in an interference, to cooperate to the best of the ability of the undersigned in the matters of preparing and executing the preliminary statement and giving and producing evidence in support thereof. The undersigned agree to perform, upon request, any affirmative acts to obtain said Letters Patent of the United States and vest in said Assignees all rights therein, whereby said Letters Patent will be held and enjoyed by said Assignees, to the full end of the term for which said Letters Patent may be granted as fully and entirely as the same would have been held and enjoyed by the undersigned if this assignment has not been made.

And for said consideration, the undersigned hereby assign to said Assignee the entire right, title and interest in said invention or improvements for all foreign countries, including all priority rights, and agree to execute, at the request of said Assignees, all documents in connection with any application for foreign Letters Patent therefor.

IN WITNESS WHEREOF, I have hereunto set my hand the 31<sup>st</sup> day of October, 2006.

  
\_\_\_\_\_  
Roger D. Chamberlain

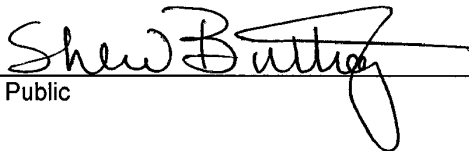
STATE OF Missouri

:  
: SS  
:

COUNTY OF St. Louis

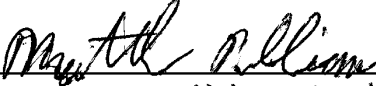
On this 31<sup>st</sup> day of October, 2006 before me, a Notary Public, within and for the County of St. Louis, State of Missouri, personally appeared ROGER D. CHAMBERLAIN, to me known to be the person described in and who executed the foregoing assignment and acknowledged that he executed same as his free act and deed.

IN TESTIMONY WHEREOF, I have hereunto set my hand and seal the date and year last above written.

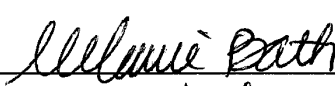
  
\_\_\_\_\_  
Notary Public

My Commission Expires: 4-4-09

WITNESS:

By:   
\_\_\_\_\_  
Printed Name: Matthew Pulliam

WITNESS:

By:   
\_\_\_\_\_  
Printed Name: Melanie Bath



IN WITNESS WHEREOF, I have hereunto set my hand the 30<sup>th</sup> day of October, 2006.

*Scott Parsons*  
Scott Parsons

STATE OF \_\_\_\_\_  
COUNTY OF \_\_\_\_\_

:  
: SS  
:

On this 30<sup>th</sup> day of October 2006 before me, a Notary Public, within and for the County of \_\_\_\_\_, State of \_\_\_\_\_, personally appeared SCOTT PARSONS, to me known to be the person described in and who executed the foregoing assignment and acknowledged that he executed same as his free act and deed.

IN TESTIMONY WHEREOF, I have hereunto set my hand and seal the date and year last above written.

*Kenya R. Lewis*  
Notary Public

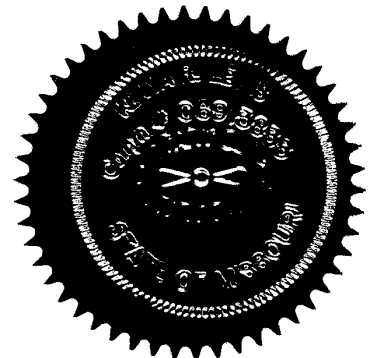
My Commission Expires: 08/08/2010

WITNESS:

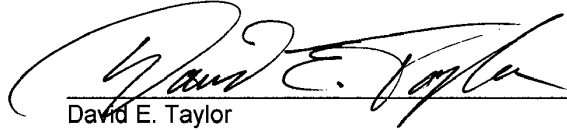
By: *Cara Lambert*  
Printed Name: Cara Lambert

WITNESS:

By: *G.A. Allswe*  
Printed Name: Gregory A. Allswe



IN WITNESS WHEREOF, I have hereunto set my hand the 30 day of October, 2006.

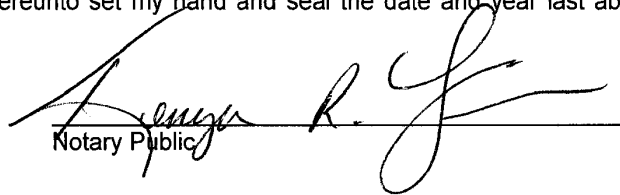
  
David E. Taylor

STATE OF  
COUNTY OF

:  
: SS  
:

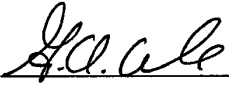
On this 30<sup>th</sup> day of October, 2006 before me, a Notary Public, within and for the County of St. Charles, State of \_\_\_\_\_, personally appeared DAVID E. TAYLOR, to me known to be the person described in and who executed the foregoing assignment and acknowledged that he executed same as his free act and deed.

IN TESTIMONY WHEREOF, I have hereunto set my hand and seal the date and year last above written.

  
Notary Public

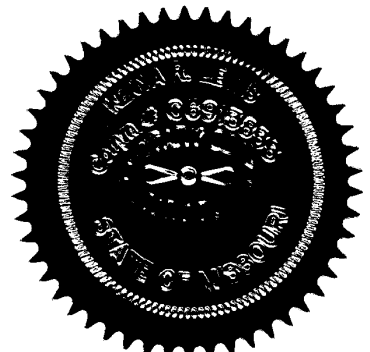
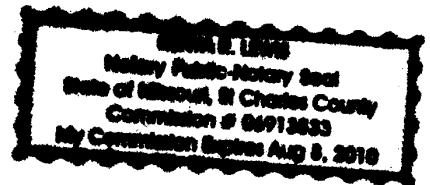
My Commission Expires: 08/08/2010

WITNESS:

By:   
Printed Name: Gregory A. Allswalter

WITNESS:

By:   
Printed Name: Cara Lambert



IN WITNESS WHEREOF, I have hereunto set my hand the 30<sup>th</sup> day of October, 2006.

Rod Arbaugh  
Rod Arbaugh

STATE OF  
COUNTY OF

:  
: SS  
:

On this 30<sup>th</sup> day of October, 2006 before me, a Notary Public, within and for the County of St. Charles, State of Missouri, personally appeared ROD ARBAUGH, to me known to be the person described in and who executed the foregoing assignment and acknowledged that he executed same as his free act and deed.

IN TESTIMONY WHEREOF, I have hereunto set my hand and seal the date and year last above written.

Kenya R. Lewis  
Notary Public

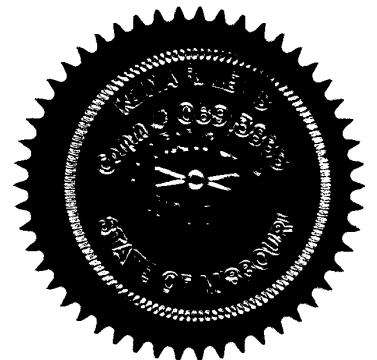
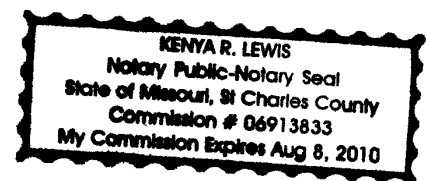
My Commission Expires: 08/08/2010

WITNESS:

By: Cara Lambert  
Printed Name: Cara Lambert

WITNESS:

By: G. A. Allswede  
Printed Name: Gregory A Allswede



IN WITNESS WHEREOF, I have hereunto set my hand the 21<sup>st</sup> day of December, 2006.

Mark A. Franklin  
Mark A. Franklin

STATE OF Missouri

COUNTY OF St. Louis

:  
: SS  
:

On this 21<sup>st</sup> day of December, 2006 before me, a Notary Public, within and for the County of St. Louis, State of Missouri, personally appeared MARK A. FRANKLIN, to me known to be the person described in and who executed the foregoing assignment and acknowledged that he executed same as his free act and deed.

IN TESTIMONY WHEREOF, I have hereunto set my hand and seal the date and year last above written.



Sheri Buttrey  
Notary Public

My Commission Expires: \_\_\_\_\_

WITNESS:

By: Cara Lambert  
Printed Name: Cara Lambert

WITNESS:

By: Melanie Bath  
Printed Name: Melanie Bath

Electronic Acknowledgement Receipt	
<b>EFS ID:</b>	1413673
<b>Application Number:</b>	60814796
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	9945
<b>Title of Invention:</b>	High speed processing of financial information using FPGA devices
<b>First Named Inventor/Applicant Name:</b>	Roger D. Chamberlain
<b>Customer Number:</b>	21888
<b>Filer:</b>	Benjamin Lowell Volk/Laurie Poss
<b>Filer Authorized By:</b>	Benjamin Lowell Volk
<b>Attorney Docket Number:</b>	44826/62269
<b>Receipt Date:</b>	03-JAN-2007
<b>Filing Date:</b>	19-JUN-2006
<b>Time Stamp:</b>	16:18:31
<b>Application Type:</b>	Provisional

### Payment information:

Submitted with Payment	no
------------------------	----

### File Listing:

Document Number	Document Description	File Name	File Size(Bytes)	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	POA.PDF	61102	no	1

### Warnings:

<b>Information:</b>					
2	Assignee showing of ownership per 37 CFR 3.73(b).	Statement.PDF	322963	no	8
<b>Warnings:</b>					
<b>Information:</b>					
<b>Total Files Size (in bytes):</b>			384065		
<p><b>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</b></p> <p><b><u>New Applications Under 35 U.S.C. 111</u></b>  If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><b><u>National Stage of an International Application under 35 U.S.C. 371</u></b>  If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p>					