Intel® QuickAssist Technology
FSB-FPGA Accelerator Architecture

Ian D. McCallum
Principal Engineer

QATS0002
Agenda

• Intel® QuickAssist® Overview
• FAP* System Architecture
• Accelerator Hardware Module (AHM)
• Accelerator Function Unit (AFU)
• Accelerator Abstraction Layer (AAL)

* FSB-FPGA Accelerator Platform
Intel® QuickAssist Technology

Encompasses Industry Hardware Solutions

Future Intel Processor Integration of Accelerators

Software Architecture Abstraction Layer and Libraries For Acceleration

Comprehensive Approach To Acceleration
Open Attach Strategy

Source: Pat Gelsinger Keynote Fall IDF 2006

1. **Open Ubiquitous Standards Based Approach**
   PCI Express* Gen1, PCI Express* Gen2, and **Geneseo** – (Extend PCI Express* Gen 2 - Joint Intel/IBM Proposal in PCI-SIG*)

2. **Enable third party FSB-FPGA Modules** – targeted for Financial Services Industry, Oil and Gas, Life Sciences, Digital Health, etc.
   FSB-FPGA Modules Targeted 4Q07/1Q08

3. **Intel® QuickAssist Technology Accelerator Abstraction Layer (AAL)** that seamlessly allows the SW to access acceleration across various technologies.

Source: Intel Internal

---

**Open Standards Based Attach Strategy**
What are FSB-FPGA Accelerators?

- FPGA Accelerator Hardware Modules (AHMs) that plug into Intel® Xeon® processor sockets
  - Attach directly to the Front Side Bus (FSB)
  - FSB provides the highest performance, lowest latency interconnect
FSB-FPGA Accelerator Hardware Module

Field Programmable Gate Arrays (FPGA) allow the development of domain-specific hardware-based, parallel algorithms that execute significantly faster than equivalent algorithm in software.

Front Side Bus (FSB) provides a high performance, low latency interconnect between AHM and CPU.

Photo source: Xilinx
Multiple AHMs

- Multiple FPGA modules can be connected in a ring topology
  - Partition complex algorithms across multiple accelerator modules
  - Higher degrees of parallelization for even higher performance
  - Connect to external I/O sources
FAP System Architecture

- **Accelerator Hardware Modules (AHM)** contains FPGA, SRAM, flash memory and control logic. AAL will support multiple AHMs from different vendors. AHMs may also be interconnected with serial links.
- **Accelerator Function Units (AFU)** implement accelerated algorithms.
- **AHM Core** logic provides FSB interface logic, low-level AHM device interface and AFU interface.
- **Workspace** memory is reserved area of system memory managed by AHM driver.
  - Memory is allocated, pinned (to prevent swapping) and mapped into application’s virtual address space. Both application and AFU can access this memory directly.
- **Accelerator Abstraction Layer (AAL)** provides generic accelerator discovery, configuration and messaging services.
- **AFU proxies** implement AFU specific message formatting and API.
- **Domain Specific Libraries** may be used to provide a more abstract interface on top of AAL.
- **Applications** access accelerator functionality through the domain specific library or directly to AAL.
Basic Accelerator Operation

AAL (incl. driver)
- Allocate Workspace
- Write operands
- Send request
  - AAL validates all pointers, adds AFU page table descriptor

Workspace
- Allocate, pin and mmap memory
- Send request
  - AAL allocates workspace memory, maps into application space and creates AFU page table

FSB CORE
- FSB Core guarantees that reads and writes are within workspace
- Get request
- AFU blocks waiting for message
- AFU unblocks, gets request
- Read operand(s)
- Write results
- Send response
- NB: AHM participates in FSB coherency protocol so “correct” memory contents are always read.

Application/Library
- Allocate Workspace
- Write operands
- Send request

AFU
- Execute

Zero-Copy Programming Model
AHM Architecture

- AHM
  - FPGA
    - SRAM
    - AHM CORE
    - MULTI-GIGABIT TRANSCEIVER
    - AFU
    - SYSTEM CONTROL AND BOOT LOGIC
    - FLASH MEMORY
  - JTAG
  - IPIP INTERFACE (FSB)

- MGTTx
- MGTRx

Intel Developer FORUM
Shared Memory: SMP

- In a typical SMP system, all CPUs can see system memory, but they can’t see each other because they don’t occupy address space.
- CPUs usually communicate through shared memory and signal each other using Inter-Processor Interrupts (IPIs).
Shared Memory: CPU+AHM

- A similar technique is used to communicate with FSB-attached AHMs
- The AHM device driver allocates (and pins) system memory for AHM device registers, command/response queues and shared workspaces
- The physical address of the device register memory area is sent to the AHM
- CPU communicates with AHM through this memory-based interface
AHM Core Architecture
AFU Interface

- One pair of Command and Data FIFOs in each direction
- Separate CSR and Bulk data transfer interfaces
- Command FIFO holds Cmd_Hdr indicating the type of transfer
- Data FIFO holds data corresponding to each Cmd_Hdr

Benefits
- High performance
- Use FPGA vendor FIFO macro implementation
- Simple FIFO interface can also be used to cross clock boundaries
- Variable burst size, 64B to 4MB
- Supports bursting with/without wait states
- Receiver controls dataflow
AFU Slave Mode Operation

Low level AFU command includes list of physical addresses and lengths for both input and output buffers. AFU is unaware of memory addressing.

AFU must process data in "application determined" order.

Write data to workspace
Write command to AFU queue
Send signal to AHM (IPI)
Read command from AFU queue
Loop
Read input from workspace
Write output to workspace
Write response to AFU queue
Send signal to CPU (IRQ)
Read response from AFU queue

Write command
Push input to AFU
Pull output from AFU
Process Data
Write response
Multiple languages will be available for developing FPGA accelerator algorithms:
- VHDL / Verilog
- High Level Languages (C, C++, Java)
- Matlab

Intel is working with a variety of third party tool vendors to provide FPGA software development kits.

Flexible Design Choices
What is AAL?

- AAL provides a uniform set of platform level services for using FPGA accelerators on Intel platforms
  - Independent of attach technology (ie: FSB, PCIe*, etc.)
  - Independent of the acceleration workload
  - Defines a common programming model
  - Specifies a common presentation of the services using existing interconnect interfaces
    - Tightly coupled accelerators: FSB
    - Loosely coupled accelerators: PCIe*
AAL Services

• Installation and configuration of accelerators on the platform
  - Standard interface for installers to register the packages they are installing
  - Standard interface for applications / libraries to query, enumerate, find and load installed packages

• Communication between applications and accelerators
  - Asynchronous programming model with event dispatcher and delivery: optimized for parallel task processing
  - Protection of shared resources allowing multiple applications to use one accelerator: support for multiple threads
  - Dynamic binding to acceleration packages: provides maximum flexibility without changing framework
AAL Interfaces

• Installation and Configuration
  - AFU Registration Interface (IRegistrar)
    ▪ Used by installer to register AFU packages
    ▪ Administrative privilege required
  - AAL System Interface (ISystem)
    ▪ Called once per process to initialize AAL
  - AAL Factory Interface (IFactory)
    ▪ Called to create AFU proxy object instances

• Communication
  - AAL Event Delivery Service (IEDS)
    ▪ Used to deliver events and support different threading models
  - AIA Interface (IAFU)
    ▪ Implemented by AFU proxy to support data exchange with AHM
    ▪ User level privilege since device driver validates all workspace memory references

Using AAL protects your software investment
Summary

• Intel is defining Intel® QuickAssist technology and enabling FPGA, tool and hardware vendors, to provide customers with a complete solution for integrating FSB-FPGA acceleration into their applications
  - Comprehensive Approach To Acceleration
  - Flexible Hardware Configuration
  - Zero-Copy Programming Model
  - Simple AFU Interface, flexible design choices
  - Using AAL protects your software investment
Additional sources of information on this topic:

- Other Sessions / Chalk Talks / Labs:
  - QATS004 - Geneseo and Intel® QuickAssist Technology Architecture Overview
  - QATS003 - Accelerator Exoskeleton: Intel® Architecture Look and Feel for Heterogeneous Cores
  - QATS001 - Intel® Embedded Processor for 2008 (Tolapai) SoC Architecture Overview
  - QATL001 - PCIe 2.0 Interop Lab
  - QATC001 - Geneseo and Intel® QuickAssist Technology Architecture Chalk Talk by Intel Fellow Ajay Bhatt
  - QATP001 - Industry Panel: Trends and challenges ahead for accelerator usage and growth

- Accelerators in Action: Visit the I/O & Application Acceleration Community in the showcase to see technology demonstrations from Intel and other industry-leading companies


This Session presentation (PDF) is available from www.intel.com/idf.
Please fill out the Session Evaluation Form for your chance to win a $500 Gift card!

How?

• Use your IDF Flash Drive
• Go to an IDF Internet Station
• Go to www.Intel.com/go/myidfeval

There will be daily drawings for Gift cards – The more evaluations you fill out the more chances to win!

See drawing terms and condition in Program Guide for more information including alternative means of entry.
Risk Factors

This presentation contains forward-looking statements. All statements made that are not historical facts are subject to a number of risks and uncertainties, and actual results may differ materially. Please refer to our most recent Earnings Release and our most recent Form 10-Q or 10-K filing available on our website for more information on the risk factors that could cause actual results to differ.
Legal Disclaimer

• INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL’S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL® PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.

• Intel may make changes to specifications and product descriptions at any time, without notice.
• All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.
• Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.
• Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.
• Intel, Intel Inside, Intel QuickAssist and the Intel logo are trademarks of Intel Corporation in the United States and other countries.
• *Other names and brands may be claimed as the property of others.
• Copyright © 2007 Intel Corporation.
MULTIPLY YOUR INNOVATIONS
AND MAXIMIZE YOUR POTENTIAL.