The New Implementation methodology of FPGA

Dylan Wang
DSP Builder System Level Design Flow

Development
- System Level Simulation of Algorithm Model
  - Algorithm-level Modeling
    - MATLAB/Simulink

Implementation
- RTL Implementation
  - Synthesis, Place & Route, RTL Simulation
    - Quartus II, ModelSim

Verification
- System Level Verification of Hardware Implementation
  - System-level Verification
    - Altera Development Kits

MATLAB/Simulink
Quartus II, ModelSim
Altera Development Kits
DSP Builder System Level Design Flow

Development
- System Level Simulation of Algorithm Model

Implementation
- RTL Implementation
- RTL Simulation

Verification
- System Level Verification of Hardware Implementation

Algorithm-level Modeling
- MATLAB/Simulink

Synthesis, Place & Route, RTL Simulation
- Quartus II, ModelSim

System-level Verification
- Altera Development Kits
Altera Blockset Libraries

- **DSP Builder Advanced Blockset**
  - is **High Level Synthesis** Design
  - Constraint driven design
  - Abstracted, generic build blocks
  - Single data path logic system clock
  - Automatic pipelining and register balancing
  - High data rate support
  - Floating point support
  - Tool creates the optimized h/w implementation

- **DSP Builder Standard Blockset**
  - is **WYSIWYG** design
  - Structural design
  - Hardware-like building blocks
  - Multiple clock domain design
  - User HDL and DSP IP import
  - Hardware Co-simulation
  - Enables fine-grain control of h/w implementation
Components in system use different interfaces to communicate (some standard, some non-standard)

Typical system requires significant engineering work to design custom interface logic

Integrating design blocks and intellectual property (IP) is tedious and error-prone
Automatic Interconnect Generation

- Avoids error-prone integration
- Saves development time with automatic logic & HDL generation
- Enables you to focus on value-add blocks

Qsys improves productivity by automatically generating the system interconnect logic.
Introducing Qsys
OpenCL (Open Computing Language) Overview

- **Software programming model:**
  - C/C++ API for host program
  - OpenCL C for acceleration device

- **Provides increased performance with hardware acceleration**
  - CPU offload to appropriate accelerator
    - Local Memory
    - Explicit Parallelism
      - Task (SMT)
      - Data (SPMD)

- **Open, royalty-free, standard**
  - Managed by Khronos Group
  - Altera active member
  - Conformance requirements
    - V1.0 is current reference
    - V2.0 is current release
  - [http://www.khronos.org](http://www.khronos.org)
Heterogeneous Platform Model

OpenCL Platform Model:
- Host Memory
- Global Memory
- Host
- Device
- Device

Example Platform:
- x86
- PCIe

© 2014 Altera Corporation—Confidential
main() {
    read_data( ... );
    manipulate( ... );
    clEnqueueWriteBuffer( ... );
    clEnqueueNDRange(..., sum,...);
    clEnqueueReadBuffer( ... );
    display_result( ... );
}

__kernel void sum
(__global float *a,
 __global float *b,
 __global float *y)
{
    int gid = get_global_id(0);
    y[gid] = a[gid] + b[gid];
}
What happens in future

- Frame work + core processing
- Auto HDL code generating
- Design based on prototype tools and diagram
- Have the chip design without deeply understanding of the device
Thank You