Fractal Video Compression in OpenCL:
An Evaluation of CPUs, GPUs, and FPGAs as Acceleration Platforms

Doris Chen, Deshanand Singh

Jan 24th, 2013

© 2013 Altera Corporation—Public

30 YEARS LEADERSHIP INNOVATION COMMUNITY

ALTERA® MEASURABLE ADVANTAGE™

Platform Evaluation Challenges

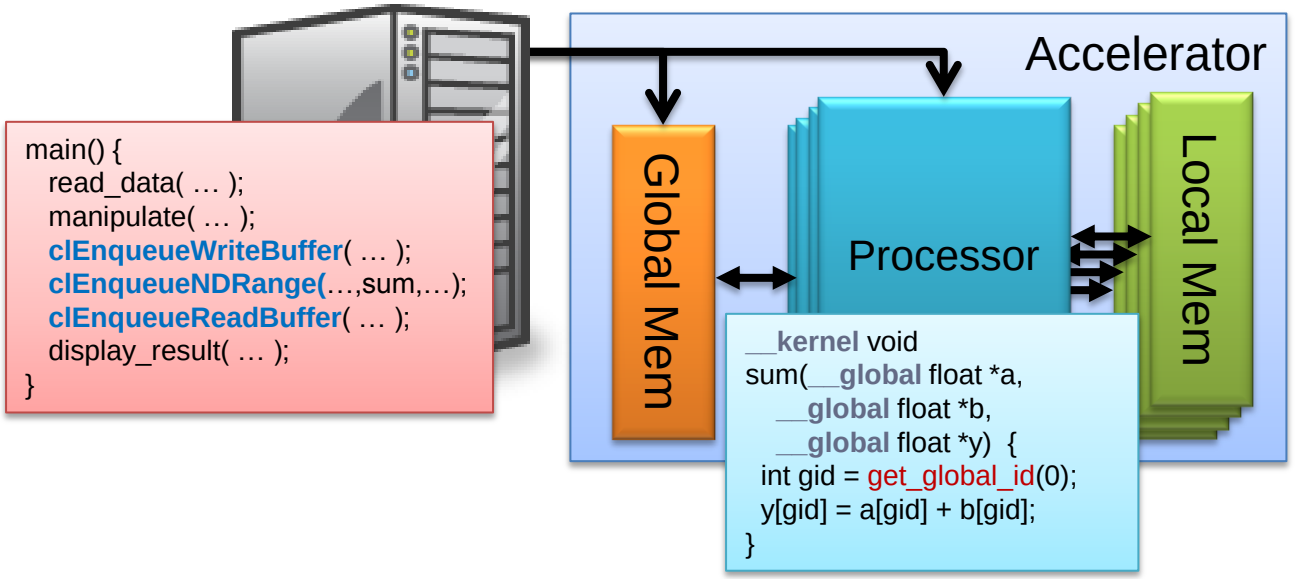
- **Conducting a fair platform evaluation is difficult**
  - Platforms have different programming models
    - CPUs – C/C++
    - GPU - vendor-specific languages (CUDA)
    - FPGAs – RTL languages (Verilog, VHDL)
- **Hard to predict behaviour ahead of time without actual implementation**
- **Designers often select the platform based on device specs**
  - Need to consider the actual functionality needed

© 2013 Altera Corporation—Public

ALTERA
MEASURABLE ADVANTAGE™

OpenCL Standard

- **A platform-independent standard**
  - Target Host + Accelerator applications
- **Data parallelism is explicitly specified**
  - **Host** – manages data and control flow
  - **Kernel** – highly parallel section to be accelerated (multi-core CPU, GPU, or FPGA)
- **Same code can be easily targeted to different platforms for performance comparison**



© 2013 Altera Corporation—Public

Objective

- **A programming model study using OpenCL**
- **Implement fractal compression in OpenCL**
  - An video encoding algorithm
- **Code is ported to and optimized for multi-core CPUs, GPUs and FPGAs**
  - Introduce Altera's OpenCL-to-FPGA Compiler
- **Compare performance between multi-core CPUs, GPUs, and FPGAs**

© 2013 Altera Corporation—**Public**

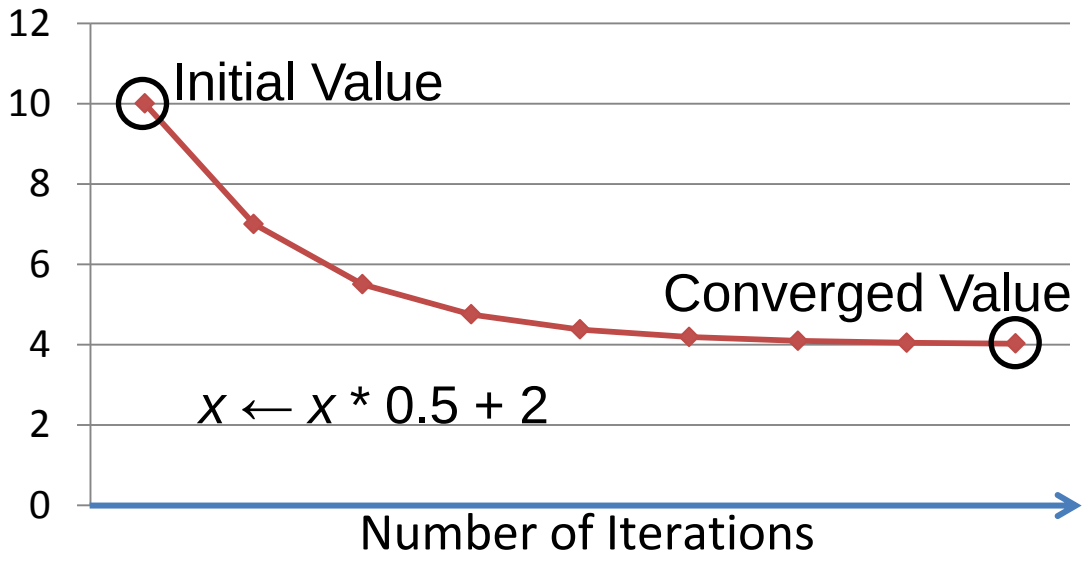
ALTERA®
MEASURABLE ADVANTAGE™

Iterated Functions for Fractal Compression

- **Based on the theory of iterated function systems (IFS)**
- **Consider a pixel value of 4**
  - Can be expressed as the recurrence relation:

$$x \leftarrow x * 0.5 + 2$$

  - Can be shown that the relation resolves to 4 regardless of initial value
- **We can represent the pixel with 2 values {0.5, 2}**



© 2013 Altera Corporation—Public

Image Compression

- **Express a N x N image as a 1-D vector**

$$P = \{I(0,0), I(0,1), \ldots, I(N-1, N-1)\}$$

- **Find transformation matrix A, and vector C such that**

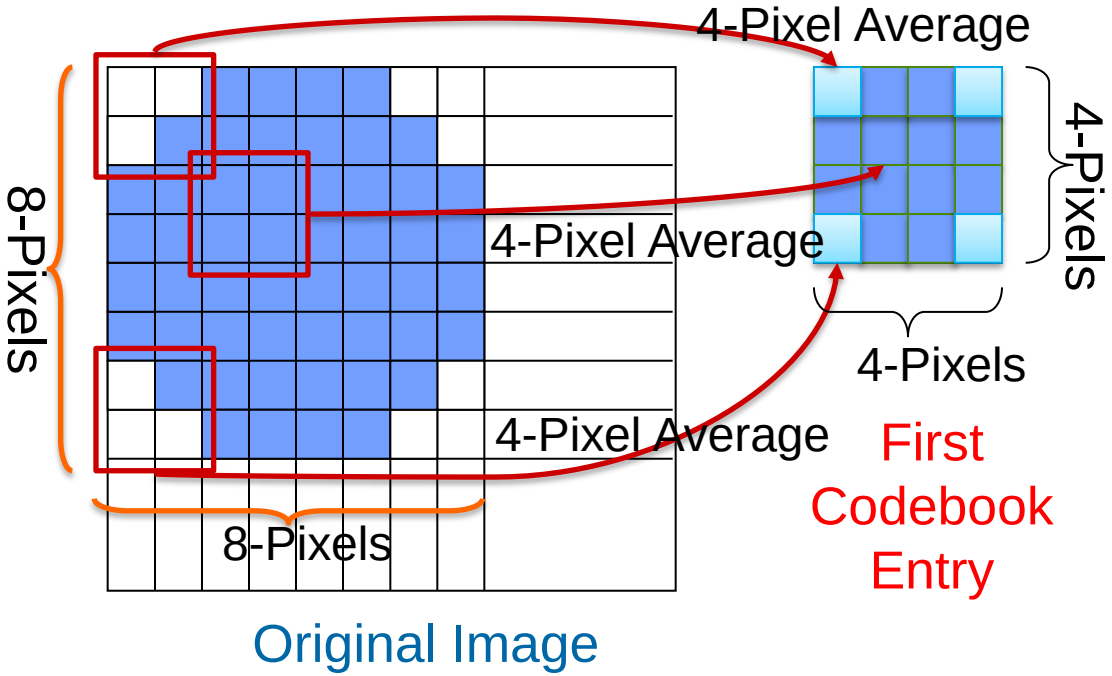
$$P \leftarrow AP + C$$

- **We would recursively apply this to arrive at the image**
  - Do not need the original image
  - Regenerate the image from random data with A and C
- **Not quite useful when performing image compression!**
  - Instead of N^2 pixel values, have a N^2 x N^2 matrix and a N^2 vector !!

© 2013 Altera Corporation—**Public**

Fractal Image Compression

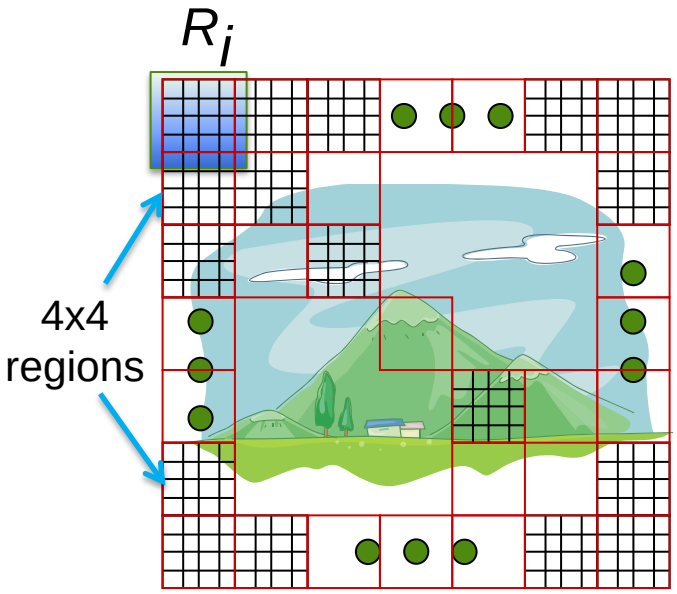
- **Creation of a codebook**
- **Take 8x8 regions of the image, and compress into 4x4**
  - Each 4x4 pixel is an average of a 2x2 region
- **Generates a library of codebook images**



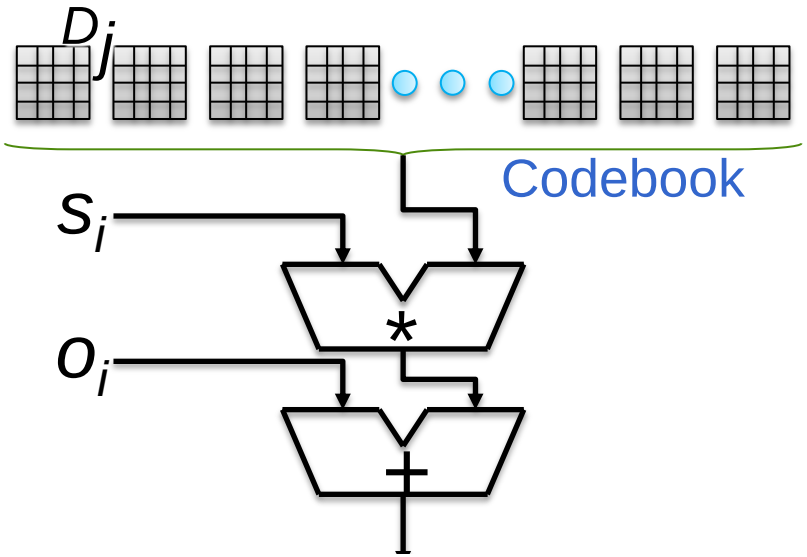
© 2013 Altera Corporation—Public

Using the Codebook

- **Codebook library help generate matrix A and vector C**
- **For each region R_i, find the codebook entry that best minimizes error**
  - After applying a scale factor s_i, and an offset o_i
- **Compute the summation of absolute differences (SAD)**
  - Compare R_i with all Dj in the codebook



Original Image

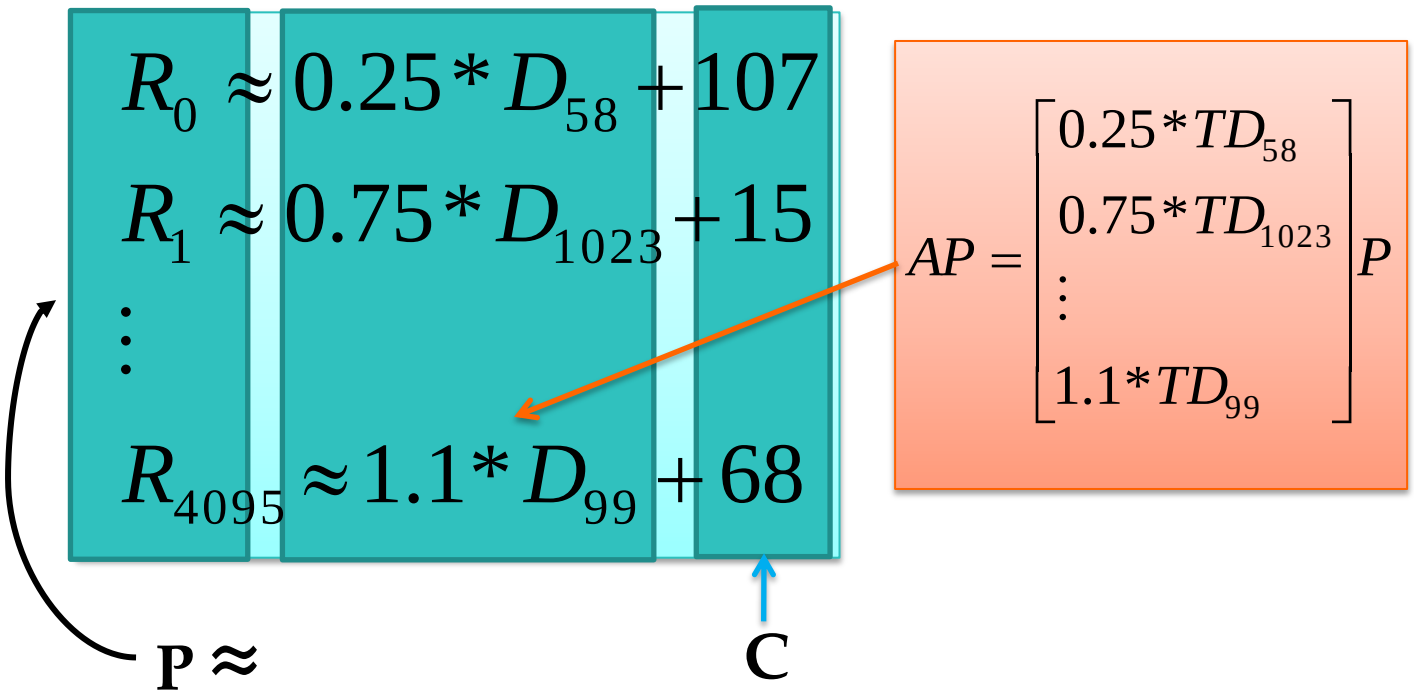


$$\min_{D_j \in Codebook} \min_{s_i, o_i} ||R_i - (s_i * D_j + o_i)||$$

© 2013 Altera Corporation—Public

Using the Codebook

- After finding the best approximation for each R_i, we may have a set of equations such as the following:



© 2013 Altera Corporation—Public

ALTERA
MEASURABLE ADVANTAGE™

Fractal Code

- **Each R_i can be expressed as:**

$$R_i = s_i * D_j + o_i$$

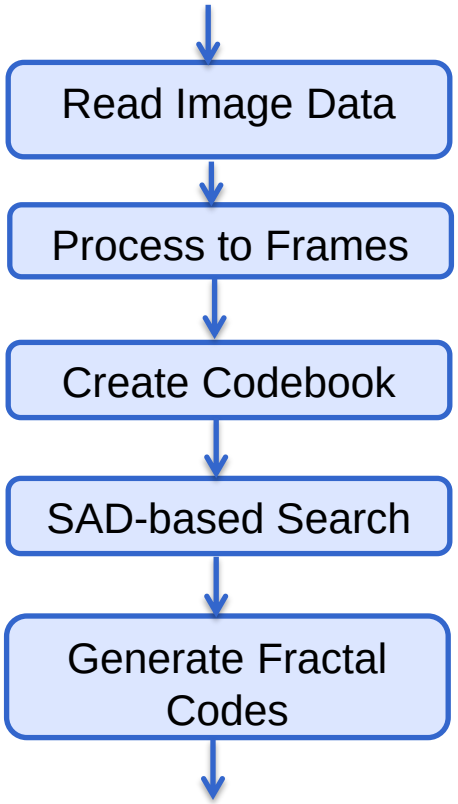
- **Transmit 3 values for each 4x4 region**
  - Best matching codebook entry *j*, scale factor s_i, and offset o_i
- **This forms the fractal code for the image**
  - Compression achieved due to transmission of a small number of coefficients

© 2013 Altera Corporation—**Public**

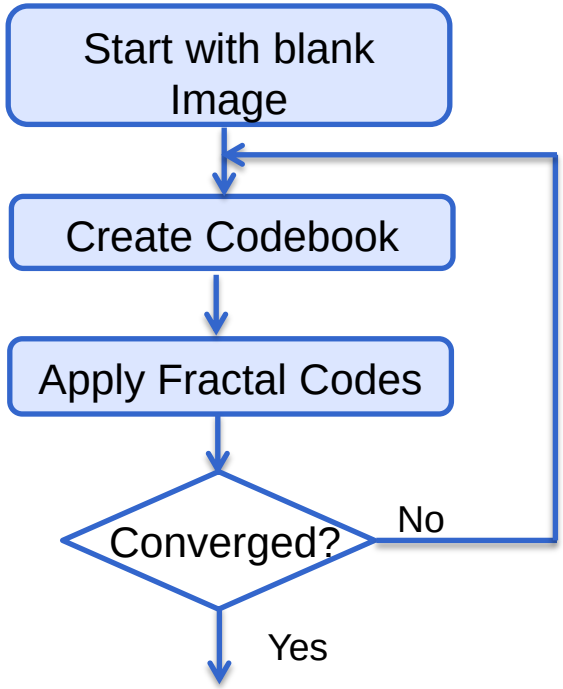
ALTERA
MEASURABLE ADVANTAGE™

Image Encoding / Decoding

Encoding



Decoding



convergence typically occurs within 4-5 iterations

© 2013 Altera Corporation—**Public**

Decoding Example

Original Image

Iteration 6

SNR = 24.4993 dB

Fractal Video Encoding General Approach

- **Encode *new* frames using the codebook of the *original* frame**
- **Cross-coding experiment showed average PSNR loss of only ~1dB**
  - Use the codebook of one image to encode another
  - Sequential frames are generally similar in nature (less diverse than random images)

Generate codebook *c1* from frame *f1*

Encode *f1* using *c1*

Encode *f2* using *c1*

Send fractal code

Regenerate f1 from code (f1')

Generate codebook *c1'* from *f1'*

Regenerate f2 from code using c1' → f2'

© 2013 Altera Corporation—Public

Optimized Kernel Code

```
__kernel void compute_fractal_code(  short* currImage, short* codeBook, ...) {
 short myImage[16], centry[16];

 int average = 0;
int imageOffset = get_global_id(0) * 16;
for (each pixel i in region) {  //loop is unrolled
  short val = currImage[imageOffset+i];
  average += val;
  myImage[i] = val; }
average >>= 4; //divide by 16 to get average
for (each pixel in region) //loop is unrolled
  myImage[i] -= average;

 ushort best_sad = 16 * 256 * 2;
 int bestCodebookEntry = -1, bestScale = -1;
for (each codebook entry icode ) {
    for (i=0; i<16; i++) //loop is unrolled
        centry[i] = codeBook[icode*16+i];
    for (each scale factor sFac) {
        ushort sad = 0;
        for (i=0; i<16; i++) //loop is unrolled
           sad += abs(sFac * centry[i] - myImage[i]);
        if (sad < best_sad) {
           best_sad = sad;
           bestCodebookEntry = icode;
           bestScale = sFac;
}}}
```

Compute average (offset)

Compare against all codebook entries

7 scale factors
{1.0, 0.75, 0.5, 0.25, -0.25, -0.5, -0.75, -1.0}

© 2013 Altera Corporation—Public

Platforms Evaluated

- **Used the latest platforms available at time of writing**

Test Platform	Representative	Process	Memory Bandwidth	Cache Size	Board Power
Multi-Core CPU	Intel Xeon W3690	32nm	32 GB/s	12 MB	130W
GPU	NVIDIA Fermi C2075	40nm	144 GB/s	768 KB	215W
FPGA	Altera Stratix IV 530	40nm	12.8 GB/s	none	21W
FPGA	Altera Stratix V 5SGXA7	28nm	25.6 GB/s	none	25W

- **Used publicly available video sequences**
  - Full-color 704x576 videos
  - SNRs > 30dB indicate a high quality result
  - Can achieve ~9.7x compression for Y plane
    - Higher for Cr and Cb planes

Benchmark	SNR (Y)	SNR (Cb)	SNR (Cr)
city	30.9	43.3	46.6
crew	34.3	43.0	40.3
harbour	32.4	43.3	45.1
soccer	33.3	42.2	44.7
ice	37.0	44.2	46.7

© 2013 Altera Corporation—Public

ALTERA
MEASURABLE ADVANTAGE™

Multi-Core CPU Results

- **Intel Xeon W3690**
  - 6 cores running at 3.46GHz, plus hyperthreading
  - 12 MB on-chip cache
- **Used Intel OpenCL SDK**
  - No communication overhead since kernels run on same device as host
  - Kernel time defined as the average kernel runtime per frame processed

Variant	Kernel Time(ms)	Transfer Time(ms)	Overall FPS
Serial Code	1639	0	0.60
OpenCL (float-pt)	**196.1**	0	**4.6**
OpenCL (fixed-pt)	325.9	0	2.8

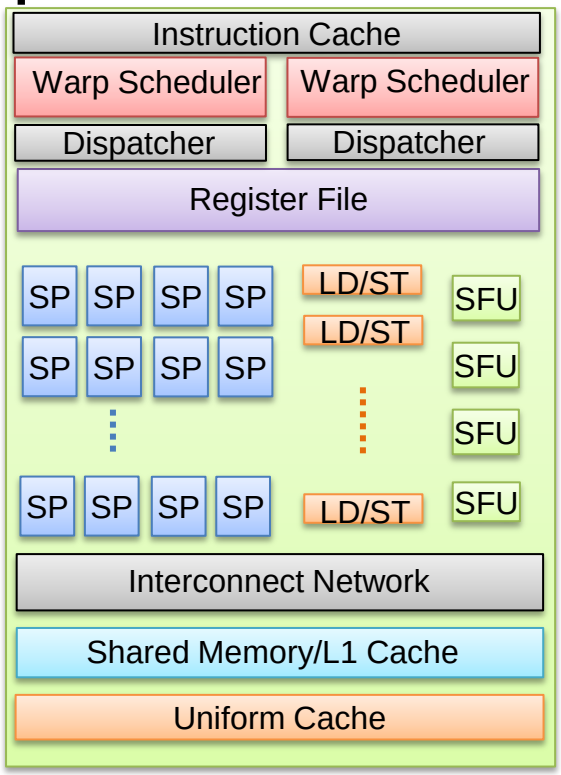
- **8x improvement when parallelized**
  - Run 1 thread per 4x4 region (no inter-thread communication)
- **Fixed-point implementation actually hurts performance**
  - The number of instructions increased

© 2013 Altera Corporation—**Public**

ALTERA
MEASURABLE ADVANTAGE™

GPU Architecture

- **Highly optimized processors known for graphics computation**
- **Designed to maximize application throughput**
- **High memory bandwidth (144GB/s)**
- **An array of compute units**
  - Streaming Multiprocessor
- **Process tens of thousands of threads in parallel**
  - Hardware support for context switching between groups of threads
- **Programmed using CUDA or NVIDIA OpenCL SDK**



Fermi GPU Streaming Multiprocessor

© 2013 Altera Corporation—Public

GPU Results

- **Fractal encoding algorithm seems to be a good fit for GPU**
  - Works on small independent regions at a time (1 thread per 4x4 region)
  - GPU can launch thousands of threads in parallel for each region
  - Low demand for shared memory or local registers
- **Specify *workgroups* (W) due to GPU hierarchy**
  - A group of threads that cooperate to solve a sub-problem
  - All threads in the workgroup run on the same SMP (CUDA core)
- **Tradeoff in workgroup size**
  - More threads → Allows for a greater ability for the GPU to hide high-latency operations
  - More threads → Greater demand on shared resources (eg. Registers)

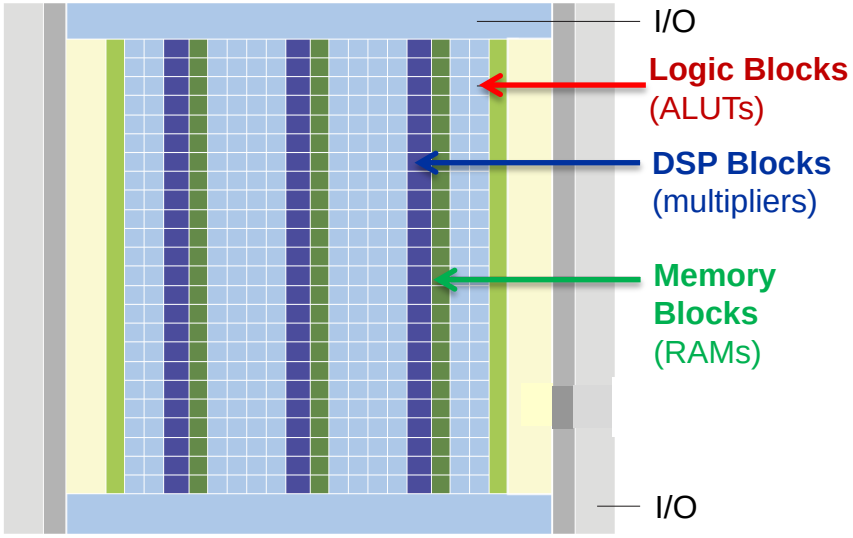
Variant	Kernel Time(ms)	Transfer Time(ms)	Overall FPS
OpenCL(W=16)	10.79	3.1	39.1
OpenCL(W=32)	6.15	3.1	50.1
OpenCL(W=64)	**5.17**	3.1	**53.1**
OpenCL(W=128)	5.28	3.1	52.8
OpenCL(W=64,fixed-pt)	16.62	3.1	31.0

© 2013 Altera Corporation—**Public**

ALTERA
MEASURABLE ADVANTAGE™

FPGAs as Accelerators

- **An array of programmable logic connected with a grid of programmable routing wires**
- **Flexible, able to implement a customized datapath**
- **Much lower power in comparison to CPUs and GPUs**
- **Memory bandwidth generally much lower**
  - Depends on board design
- **Traditionally implemented using hardware description languages**
  - Intimate knowledge of device architecture and cycle accuracy required
  - Can be a challenge for HPC adoption



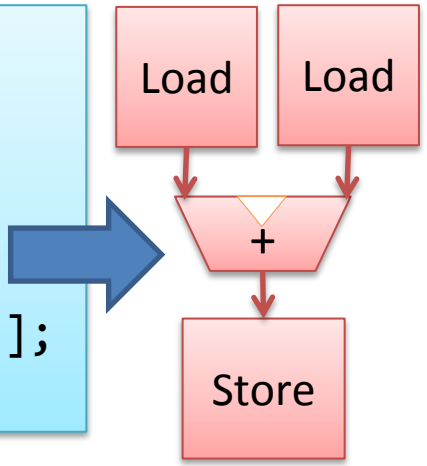
© 2013 Altera Corporation—Public

Altera OpenCL SDK

- **A high-level synthesis tool that compiles OpenCL code to HDL for FPGA implementation**

- **Translates the kernel to hardware by creating a circuit to implement each operation**

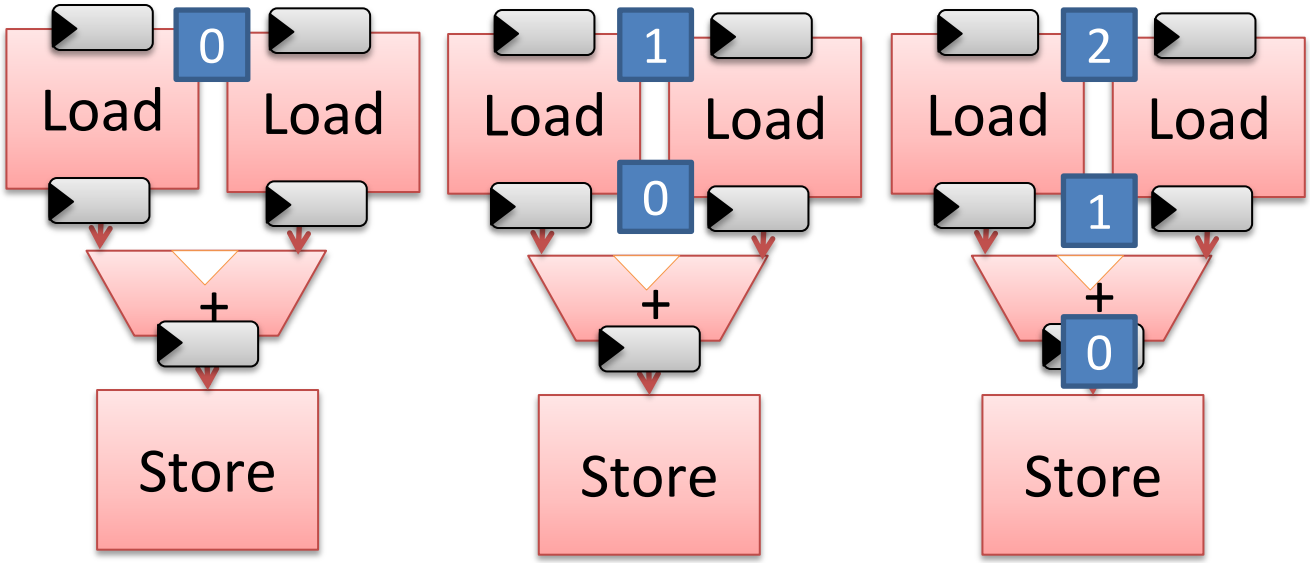
```
__kernel void
sum(__global const float *a,
__global const float *b,
__global float *answer) {
  int xid = get_global_id(0);
  answer[xid] = a[xid] + b[xid];
}
```



© 2013 Altera Corporation—Public

Pipeline Parallelism

- **Run threads in parallel using the same hardware**
- **Each parallel thread is associated with an ID**
  - Indicates the subset of data it operates on



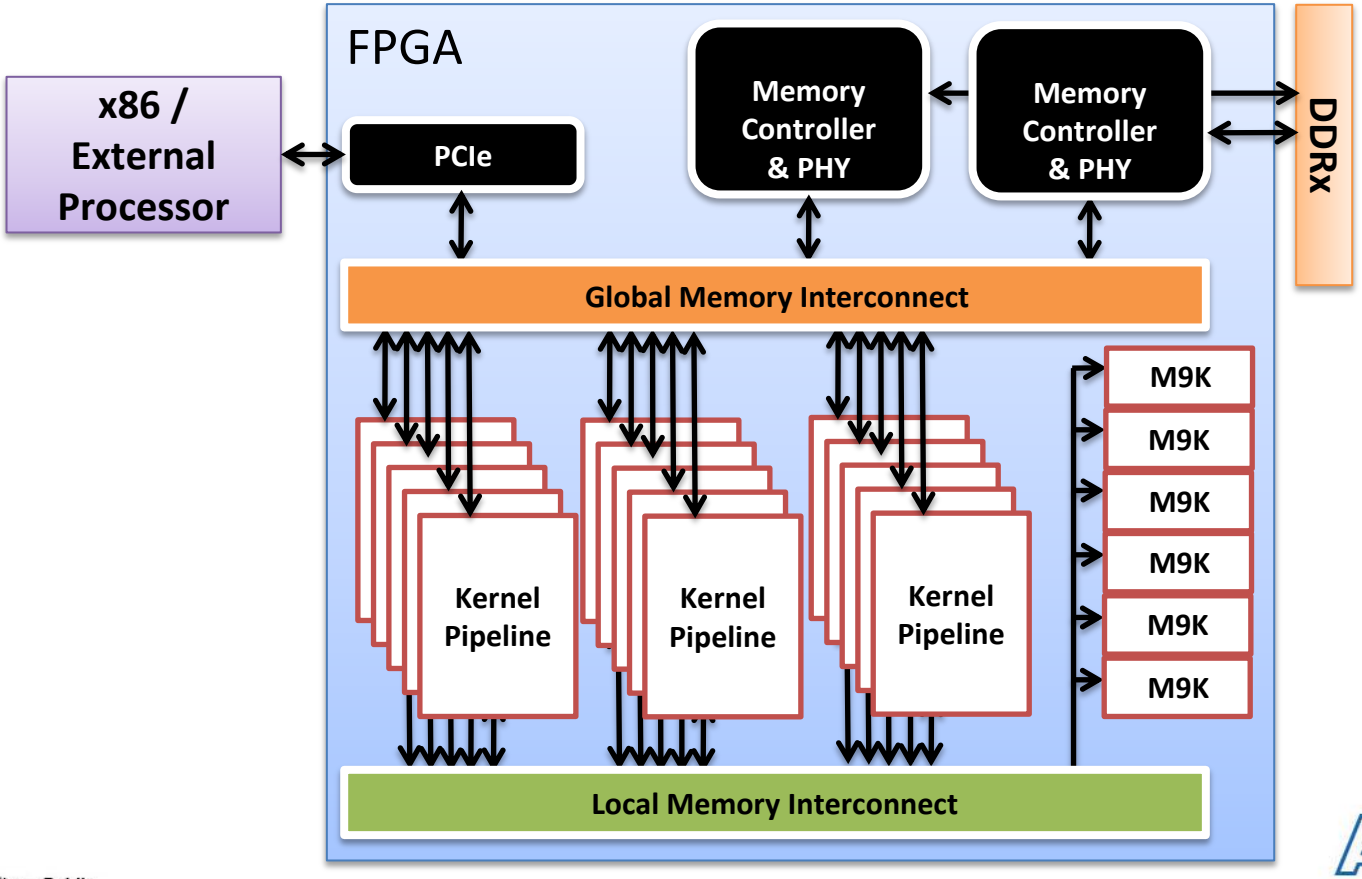
(a) First Cycle (b) Second Cycle (c) Third Cycle

© 2013 Altera Corporation—Public

ALTERA
MEASURABLE ADVANTAGE™

System Solution

- Compiler creates interfaces to external and internal memory
- Automatically timing closed



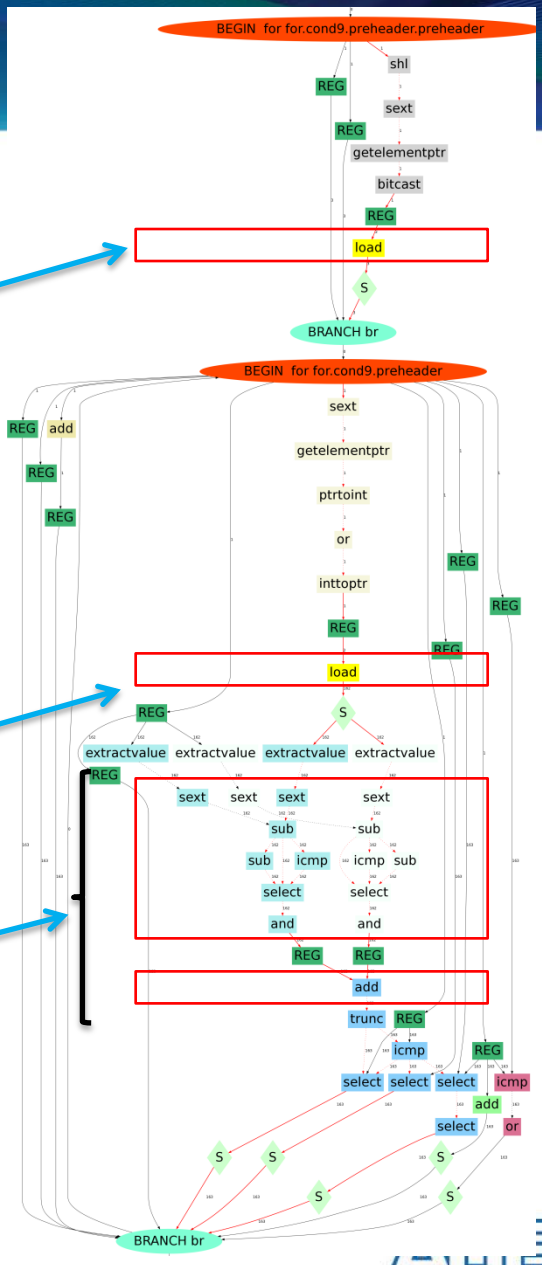
© 2013 Altera Corporation—Public

Fractal Video Kernel

```
__kernel void compute_fractal_code(  short* currImage, short* codeBook, ...) {
  short myImage[16], centry[16];

  int average = 0;
int imageOffset = get_global_id(0) * 16;
for (each pixel i in region) {  //loop is unrolled
  short val = currImage[imageOffset+i];
  average += val;
  myImage[i] = val; }
average >>= 4; //divide by 16 to get average
for (each pixel in region) //loop is unrolled
  myImage[i] -= average;

  ushort best_sad = 16 * 256 * 2;
  int bestCodebookEntry = -1, bestScale = -1;
for (each codebook entry icode ) {
    for (i=0; i<16; i++) //loop is unrolled
        centry[i] = codeBook[icode*16+i];
    for (each scale factor sFac) {
        ushort sad = 0;
        for (i=0; i<16; i++) //loop is unrolled
           sad += abs(sFac * centry[i] - myImage[i]);
        if (sad < best_sad) {
           best_sad = sad;
           bestCodebookEntry = icode;
           bestScale = sFac;
}}}
```



© 2013 Altera Corporation—Public

FPGA Results

- **Used Stratix IV and V FPGAs**
  - Easy to retarget the OpenCL code; no code change required
- **Because of pipeline parallelism, performance increases with more copies of pipeline**
  - Control this with loop unrolling

Variant	Kernel Time(ms)	Transfer Time(ms)	Overall FPS
Stratix IV			
OpenCL(U=24,fixed-pt)	**2.0**	2.2	**70.9**
Stratix V			
OpenCL(U=1,fixed-pt)	20.20	1.9	28.4
OpenCL(U=24,fixed-pt)	**1.72**	1.9	**74.4**
OpenCL(U=2,float-pt)	11.84	1.9	38.8

- **Fixed-point computations improve performance significantly**

© 2013 Altera Corporation—**Public**

ALTERA
MEASURABLE ADVANTAGE™

Summary

- **Performed platform evaluation using OpenCL**
- **Implemented fractal encoding and map it efficiently for multi-core CPUs, GPUs, and FPGAs**

Platform	Kernel Runtime	FPS	Board Power
Multi-Core CPU (Intel Xeon W3690)	196.1	4.6	130 W
GPU (NVIDIA Fermi 2075)	5.17	53.1	215 W
FPGA (Altera Stratix V 5SGXA7)	1.72	74.4	25 W

- **Showed that core computation can be 3x faster on FPGA vs. GPU with 8x less power**
- **Using a High Level Synthesis tool can dramatically reduce the time required for FPGA implementation**

© 2013 Altera Corporation—Public

ALTERA
MEASURABLE ADVANTAGE™

Thank You

© 2013 Altera Corporation—**Public**

ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/legal.

Backup

© 2013 Altera Corporation—Public

Objective

- **A programming model study using OpenCL**
- **Implement fractal compression in OpenCL**
  - An video encoding algorithm based on iterated function systems (IFS)
- **Code is ported to and optimized for multi-core CPUs, GPUs and FPGAs**
- **Introduce Altera's OpenCL-to-FPGA Compiler**
- **Performance comparisons**
  - Between multi-core CPUs, GPUs, and FPGAs
  - Between OpenCL and hand-coded RTL implementations

© 2013 Altera Corporation—**Public**

Fractal Video Compression

- **Encode *new* frames using the codebook of the *original* frame**
- **Cross-coding experiment**
  - Use the codebook of one image to encode another

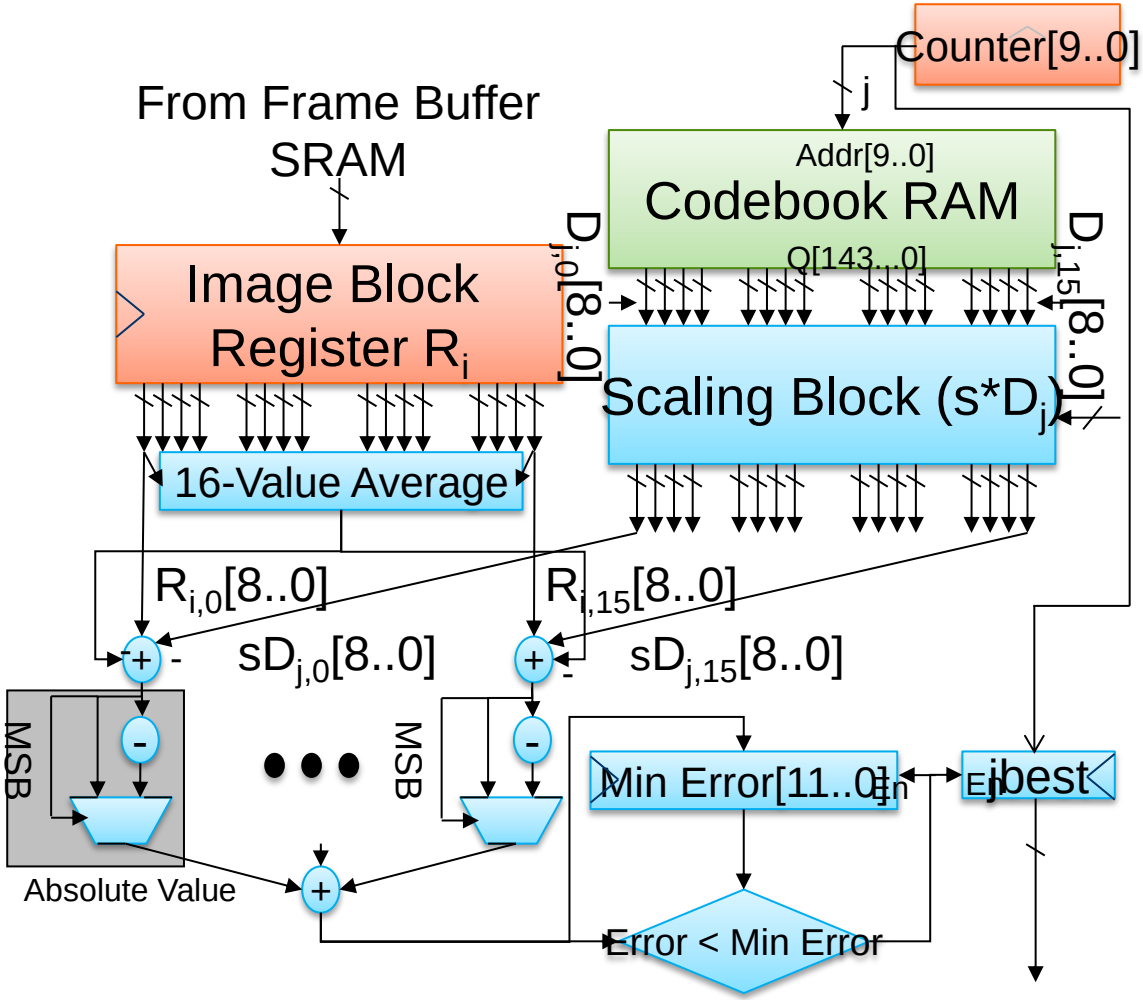
	Aerial	Airplane	Balloon	Girl	Lenna	Mandrill	Parrots	Pepper	Sailboat	Couple	Milkdrop
Aerial	**27.5**	25.7	34.1	30.8	27.6	24.6	27.9	26.5	27.1	30.5	30.0
Airplane	27.0	**27.3**	34.0	31.0	28.2	24.3	28.5	27.1	27.4	30.3	31.7
Balloon	26.9	25.1	**34.9**	30.2	27.4	24.1	27.5	25.8	26.5	30.3	29.6
Girl	27.0	25.1	34.6	**32.1**	28.0	24.2	27.6	26.4	26.9	30.9	29.7
Lenna	26.9	26.0	34.6	31.3	**29.5**	24.3	28.4	27.1	27.2	31.1	31.1
Mandrill	27.0	25.4	34.0	30.7	27.8	**24.6**	27.8	26.6	26.8	30.9	29.6
Parrots	27.1	26.3	34.8	31.2	28.5	24.2	**28.7**	27.0	27.3	30.7	30.9
Pepper	27.4	26.6	35.0	31.9	29.1	24.6	28.7	**28.7**	28.3	31.8	31.6
Sailboat	27.5	27.0	34.8	31.6	28.5	24.7	28.7	27.5	**28.3**	31.3	31.5
couple	27.1	25.1	34.7	30.9	27.7	24.3	27.6	26.6	26.8	**31.6**	29.8
milkdrop	26.9	25.8	34.1	31.0	27.5	23.9	28.0	26.3	26.7	30.1	**31.1**
Average (dB)	26.9	25.1	34.0	30.2	27.4	23.9	27.5	25.8	26.5	30.1	29.6
Loss (dB)	-0.6	-2.3	-1.0	-1.9	-2.1	-0.7	-1.2	-2.9	-1.8	-1.5	-1.5

  - Show an average loss in PSNR of only ~1dB

© 2013 Altera Corporation—**Public**

Hand-coded RTL Implementation

- **Onchip frame buffer and codebook**



© 2013 Altera Corporation—Public

OpenCL vs. Handcoded

- **Core computation replicated for each scale factor**

	Handcoded	OpenCL SDK
Frame buffer	On-chip memory	Global memory
Codebook	On-chip memory	Global memory
Unroll	4x	24x
Development Time	>1 month	Hours

- **OpenCL design flow greatly simplifies FPGA implementation**
  - Automates external interfacing such as DDR and PCIe

© 2013 Altera Corporation—Public

ALTERA
MEASURABLE ADVANTAGE™