Summary of OpenCL 1.1

This document lists the OpenCL constructs used in this tutorial (CERN June 2011). To keep the discussion as simple as possible, we truncate the lists of parameters, types, and properties to those used in the tutorial. For a complete summary of OpenCL, download the OpenCL reference card at http://www.khronos.org/files/opencl-1-1-quick-reference-card.pdf.

I. The OpenCL Platform Layer

1.1 Querying Platform Info and Devices

cl_int **clGetPlatformIDs** (cl_uint *num_entries*, cl_platform_id **platforms*, cl_uint **num_platforms*)

cl_int **clGetPlatformInfo** (cl_platform_id *platform*, cl_platform_info *param_name*,
size_t *param_value_size*, void **param_value*, size_t **param_value_size_ret*)

param_name: CL_PLATFORM_{PROFILE, VERSION}, CL_PLATFORM_{NAME, VENDOR, EXTENSIONS}

cl_int **clGetDeviceIDs** (cl_platform_id *platform*, cl_device_type *device_type*, cl_uint *num_entries*,
cl_device_id **devices*, cl_uint **num_devices*)

device_type: CL_DEVICE_TYPE_{CPU, GPU}, CL_DEVICE_TYPE_{ACCELERATOR, DEFAULT, ALL}

cl_int **clGetDeviceInfo** (cl_device_id *device*, cl_device_info *param_name*, size_t *param_value_size*, void **param_value*,
size_t **param_value_size_ret*)

param_name:

CL_DEVICE_TYPE,	CL_DEVICE_VENDOR_ID,
CL_DEVICE_MAX_COMPUTE_UNITS,	CL_DEVICE_MAX_WORK_ITEM_{DIMENSIONS, SIZES},
CL_DEVICE_MAX_WORK_GROUP_SIZE,	CL_DEVICE_MAX_MEM_ALLOC_SIZE,
CL_DEVICE_MAX_PARAMETER_SIZE,	CL_DEVICE_GLOBAL_MEM_CACHE_{TYPE, SIZE},
CL_DEVICE_GLOBAL_MEM_CACHELINE_SIZE,	CL_DEVICE_GLOBAL_MEM_SIZE,
CL_DEVICE_LOCAL_MEM_{TYPE, SIZE},	CL_DEVICE_PROFILING_TIMER_RESOLUTION
CL_DEVICE_{NATIVE, PREFERRED}_VECTOR_WIDTH_(FLOAT, INT, CHAR}	

1.2 Contexts

cl_context **clCreateContext** (const cl_context_properties **properties*, cl_uint *num_devices*, const cl_device_id **devices*,
void (CL_CALLBACK**pfn_notify*)(const char **errinfo*, const void **private_info*, size_t *cb*, void **user_data*),
void **user_data*, cl_int **errcode_ret*)

properties: CL_CONTEXT_PLATFORM

2. The OpenCL Runtime

2.1 Command Queues

cl_command_queue **clCreateCommandQueue** (cl_context *context*, cl_device_id *device*,
cl_command_queue_properties *properties*, cl_int **errcode_ret*)

properties: CL_QUEUE_PROFILING_ENABLE, CL_QUEUE_OUT_OF_ORDER_EXEC_MODE_ ENABLE

2.2 Program Objects

cl_program **clCreateProgramWithSource** (cl_context *context*, cl_uint *count*, const char **_strings_, const size_t *_lengths_, cl_int *_errcode_ret_)

cl_int **clBuildProgram** (cl_program *program*, cl_uint *num_devices*, const cl_device_id *_device_list_, const char *_options_, void (CL_CALLBACK*_pfn_notify_)(cl_program *program,* void *_user_data_), void *_user_data_)

cl_int **clGetProgramBuildInfo** (cl_program *program*, cl_device_id *device*, cl_program_build_info *param_name*, size_t *param_value_size*, void *_param_value_, size_t *_param_value_size_ret_)

param_name: CL_PROGRAM_BUILD_{STATUS, OPTIONS, LOG}

2.3 Kernel Objects

cl_kernel **clCreateKernel** (cl_program *program*, const char *_kernel_name_, cl_int *_errcode_ret_)

cl_int **clSetKernelArg** (cl_kernel *kernel*, cl_uint *arg_index*, size_t *arg_size*, const void *_arg_value_)

cl_int **clEnqueueNDRangeKernel** (cl_command_queue *command_queue*, cl_kernel *kernel*, cl_uint *work_dim*, const size_t *_global_work_offset_, const size_t *_global_work_size_, const size_t *_local_work_size_, cl_uint *num_events_in_wait_list*, const cl_event *_event_wait_list_, cl_event *_event_)

2.4 Buffer Objects

cl_mem **clCreateBuffer** (cl_context *context*, cl_mem_flags *flags*, size_t *size*, void *_host_ptr, cl_int *_errcode_ret_)

flags for clCreateBuffer
CL_MEM_READ_WRITE, CL_MEM_{WRITE, READ}_ONLY, CL_MEM_{USE, ALLOC, COPY}_HOST_PTR

cl_int **clEnqueueReadBuffer** (cl_command_queue *command_queue*, cl_mem *buffer*, cl_bool *blocking_read*, size_t *offset*, size_t *cb*, void *_ptr, cl_uint *num_events_in_wait_list*,const cl_event *_event_wait_list_, cl_event *_event_)

cl_int **clEnqueueWriteBuffer** (cl_command_queue *command_queue*, cl_mem *buffer*, cl_bool *blocking_write*, size_t *offset*, size_t *cb*, const void *_ptr, cl_uint *num_events_in_wait_list*, const cl_event *_event_wait_list_, cl_event *_event_)

2.5 Event Objects

cl_event **clCreateUserEvent** (cl_context *context*, cl_int *_errcode_ret_)

cl_int **clSetUserEventStatus** (cl_event *event*, cl_int *execution_status*)

cl_int **clWaitForEvents** (cl_uint *num_events*, const cl_event *_event_list_)

cl_int **clGetEventInfo** (cl_event *event*, cl_event_info *param_name*, size_t *param_value_size*, void *_param_value_, size_t *_param_value_size_ret_)

param_name: CL_EVENT_COMMAND_{QUEUE, TYPE}, CL_EVENT_{CONTEXT, REFERENCE_COUNT}, CL_EVENT_COMMAND_EXECUTION_STATUS

cl_int **clGetEventProfilingInfo** (cl_event *event*, cl_profiling_info *param_name*, size_t *param_value_size*, void *_param_value_, size_t *_param_value_size_ret_)

param_name: CL_PROFILING_COMMAND_QUEUED, CL_PROFILING_COMMAND_{SUBMIT, START, END}

2.6 Synchronization

cl_int **clEnqueueMarker** (cl_command_queue *command_queue*, cl_event *_event_)

cl_int **clEnqueueWaitForEvents** (cl_command_queue *command_queue*, cl_uint *num_events*, const cl_event *_event_list_)

cl_int **clEnqueueBarrier** (cl_command_queue *command_queue*)

cl_int **clFlush** (cl_command_queue *command_queue*)

cl_int **clFinish** (cl_command_queue *command_queue*)

3. OpenCL Data Types

3.1 Built-In Scalar Data Types

OpenCL Type	API Type	Description
char	cl_char	8-bit signed
short	cl_short	16-bit signed
int	cl_int	32-bit signed
long	cl_long	64-bit signed
float	cl_float	32-bit float
size_t	--	32- or 64-bit unsigned integer
void	void	void

3.2 Built-In Vector Data Types

OpenCL Type	API Type	Description
charn	cl_char*n*	8-bit signed
shortn	cl_shortn	16-bit signed
intn	cl_intn	32-bit signed
longn	cl_longn	64-bit signed
floatn	cl_floatn	32-bit float

3.3 Other Built-In Data Types

OpenCL Type	Description
event_t	event handle

4. Special Features of the OpenCL C Programming Language

4.1 Vector Component Addressing

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
float2 v;	v.x, v.s0	v.y, v.s1														
float3 v;	v.x, v.s0	v.y, v.s1	v.z, v.s2													
float4 v;	v.x, v.s0	v.y, v.s1	v.z, v.s2	v.w, v.s3												
float8 v;	v.s0	v.s1	v.s2	v.s3	v.s4	v.s5	v.s6	v.s7								
float16 v;	v.s0	v.s1	v.s2	v.s3	v.s4	v.s5	v.s6	v.s7	v.s8	v.s9	v.sa, v.sA	v.sb, v.sB	v.sc, v.sC	v.sd, v.sD	v.se, v.sE	v.sf, v.sF

4.2 Vector Addressing Equivalencies

Numeric indices are preceded by the letter s or S, e.g.: `s1`. Swizzling, duplication, and nesting are allowed, e.g.: `v.yx`, `v.xx`, `v.lo.x`. When using `.lo` or `.hi` with a 3-component vector, the `.w` component is undefined.

	v.lo	v.hi	v.odd	v.even
float2	v.x, v.s0	v.y, v.s1	v.y, v.s1	v.x, v.s0
float3*	v.s01, v.xy	v.s23, v.zw	v.s13, v.yw	v.s02, v.xz
float4	v.s01, v.xy	v.s23, v.zw	v.s13, v.yw	v.s02, v.xz
float8	v.s0123	v.s4567	v.s1357	v.s0246
float16	v.s01234567	v.s89abcdef	v.s13579bdf	v.s02468ace

4.3 Address Space Qualifiers

__global, global	__local, local
__constant, constant	__private, private

4.4 Function Qualifiers

__kernel, kernel

__attribute__((vec_type_hint(type))) //*type defaults to int*

__attribute__((work_group_size_hint(X, Y, Z)))

__attribute__((reqd_work_group_size(X, Y, Z)))

4.5 Built-In Functions

D is dimension index.

`uint` **`get_work_dim`** `()`	Num. of dimensions in use
`size_t` **`get_global_size`** `(uint` *`D`*`)`	Num. of global work-items
`size_t` **`get_global_id`** `(uint` *`D`*`)`	Global work-item ID value
`size_t` **`get_local_size`** `(uint` *`D`*`)`	Num. of local work-items
`size_t` **`get_local_id`** `(uint` *`D`*`)`	Local work-item ID
`size_t` **`get_num_groups`** `(uint` *`D`*`)`	Num. of work-groups
`size_t` **`get_group_id`** `(uint` *`D`*`)`	Returns the work-group ID

4.6 Synchronization, Explicit Memory Fence

`flags` argument is the memory address space, set to a combination of `CLK_LOCAL_MEM_FENCE` and `CLK_GLOBAL_MEM_FENCE`.

void **barrier** (cl_mem_fence_flags *flags*)	All work-items in a work-group must execute this before any can continue
void **mem_fence** (cl_mem_fence_flags *flags*)	Orders loads and stores of a work-item executing a kernel