arm

Arm, Cambridge, UK

Handling 1000s of OpenCL builtin functions in Clang

Pierre Gondois, Joey Gouly, Sven van Haastregt

LLVM Developers' Meeting, 22–23 October 2019

OpenCL Builtins

```
float cos(float);
```

OpenCL Builtins for various data types

```
float cos(float);
```

```
double cos(double);
```

```
half cos(half);
```

OpenCL Builtins for various data types

```
float cos(float);
```

```
#ifdef cl_khr_fp64
double cos(double);
#endif
```

```
#ifdef cl_khr_fp16
half cos(half);
#endif
```


arm

OpenCL Builtins for various data types and vector sizes

```
float cos(float);
float2 cos(float2);
float3 cos(float3);
float4 cos(float4);
float8 cos(float4);
float16 cos(float16);
```

```
#ifdef cl_khr_fp64
double cos(double);
double2 cos(double2);
double3 cos(double3);
double4 cos(double4);
double8 cos(double8);
double16 cos(double16);
#endif
```

```
#ifdef cl_khr_fp16
half cos(half);
half2 cos(half2);
half3 cos(half3);
half4 cos(half4);
half8 cos(half8);
half16 cos(half16);
#endif
```


arm

Some more OpenCL Builtins

acos, acosh, acospi, asin, asinh, asinpi, atan, atanh, atanpi, cbrt, ceil, cos, cosh, cospi, erfc, erf, exp, exp2, exp10, expm1, fabs, floor, log, log2, log10, log1p, logb, rint, round, rsqrt, sin, sinh, sinpi, sqrt, tan, tanh, tanpi, tgamma, trunc, atan2, atan2pi, copysign, fdim, fmod, hypot, maxmag, minmag, nextafter, pow, powr, remainder, prefetch, wait_group_events, vload2, vload3, vload4, vload8, vload16, vstore2, vstore3, vstore4, vstore8, vstore16, async_work_group_strided_copy, shuffle, atomic_add, atomic_sub, atomic_xchg, atomic_min, atomic_max, atomic_and, atomic_or, atomic_xor, atomic_inc, atomic_dec, read_imagef, read_imagei, read_imageui, read_imageh, write_imagef, write_imagei, write_imageui, write_imageh, get_image_width, convert_char, convert_char_rte, convert_char_rtn, convert_char_rtp, convert_char_rtz, convert_char_sat, convert_char_sat_rte, convert_char_sat_rtn, convert_char_sat_rtp, convert_char_sat_rtz, convert_char2, convert_char2_rte, convert_char2_rtn, convert_char2_rtp, convert_char2_rtz, convert_char2_sat, convert_char2_sat_rte, convert_char2_sat_rtn, convert_char2_sat_rtp, convert_char2_sat_rtz, convert_char3, convert_char3_rte, convert_char3_rtn, convert_char3_rtp, convert_char3_rtz, convert_char3_sat, convert_char3_sat_rte, convert_char3_sat_rtn, convert_char3_sat_rtp, convert_char3_sat_rtz, convert_char4, convert_char4_rte, convert_char4_rtn, convert_char4_rtp, convert_char4_rtz, convert_char4_sat, convert_char4_sat_rte, convert_char4_sat_rtn, convert_char4_sat_rtp, convert_char4_sat_rtz, convert_char8, convert_char8_rte, convert_char8_rtn, convert_char8_rtp, convert_char8_rtz, convert_char8_sat, convert_char8_sat_rte, convert_char8_sat_rtn, convert_char8_sat_rtp, convert_char8_sat_rtz, convert_char16, convert_char16_rte, convert_char16_rtn, convert_char16_rtp, convert_char16_rtz, convert_char16_sat, convert_char16_sat_rte, convert_char16_sat_rtn, convert_char16_sat_rtp, convert_char16_sat_rtz, convert_int, convert_int_rte, convert_int_rtn, convert_int_rtp, convert_int_rtz, convert_int_sat, convert_int_sat_rte, convert_int_sat_rtn, convert_int_sat_rtp, convert_int_sat_rtz, convert_int2, convert_int2_rte, convert_int2_rtn, convert_int2_rtp, convert_int2_rtz, convert_int2_sat, convert_int2_sat_rte, convert_int2_sat_rtn, convert_int2_sat_rtp, convert_int2_sat_rtz, convert_int3, convert_int3_rte, convert_int3_rtn, convert_int3_rtp, convert_int3_rtz, convert_int3_sat, convert_int3_sat_rte, convert_int3_sat_rtn, convert_int3_sat_rtp, convert_int3_sat_rtz, convert_int4, convert_int4_rte, convert_int4_rtn, convert_int4_rtp, convert_int4_rtz, convert_int4_sat, convert_int4_sat_rte, convert_int4_sat_rtn, convert_int4_sat_rtp, convert_int4_sat_rtz, convert_int8, convert_int8_rte, convert_int8_rtn, convert_int8_rtp, convert_int8_rtz, convert_int8_sat, convert_int8_sat_rte, convert_int8_sat_rtn, convert_int8_sat_rtp, convert_int8_sat_rtz, convert_int16, convert_int16_rte, convert_int16_rtn, convert_int16_rtp, convert_int16_rtz, convert_int16_sat, convert_int16_sat_rte, convert_int16_sat_rtn, convert_int16_sat_rtp, convert_int16_sat_rtz, convert_uint, convert_uint_rte, convert_uint_rtn, convert_uint_rtp, convert_uint_rtz, convert_uint_sat, convert_uint_sat_rte, convert_uint_sat_rtn, convert_uint_sat_rtp, convert_uint_sat_rtz, convert_uint2, convert_uint2_rte, convert_uint2_rtn, convert_uint2_rtp, convert_uint2_rtz, convert_uint2_sat, convert_uint2_sat_rte, convert_uint2_sat_rtn, convert_uint2_sat_rtp, convert_uint2_sat_rtz, convert_uint3, convert_uint3_rte, convert_uint3_rtn, convert_uint3_rtp, convert_uint3_rtz, convert_uint3_sat, convert_uint3_sat_rte, convert_uint3_sat_rtn, convert_uint3_sat_rtp, convert_uint3_sat_rtz, convert_uint4, convert_uint4_rte, convert_uint4_rtn, convert_uint4_rtp, convert_uint4_rtz, convert_uint4_sat, convert_uint4_sat_rte, convert_uint4_sat_rtn, convert_uint4_sat_rtp, convert_uint4_sat_rtz, convert_uint8, convert_uint8_rte, convert_uint8_rtn, convert_uint8_rtp, convert_uint8_rtz, convert_uint8_sat, convert_uint8_sat_rte, convert_uint8_sat_rtn, convert_uint8_sat_rtp, convert_uint8_sat_rtz, convert_uint16, convert_uint16_rte, convert_uint16_rtn, convert_uint16_rtp, convert_uint16_rtz, convert_uint16_sat, convert_uint16_sat_rte, convert_uint16_sat_rtn, convert_uint16_sat_rtp, convert_long_sat_rte, convert_long_sat_rtn, convert_long_sat_rtp, convert_long_sat_rtz, convert_long2, convert_long2_rte, ...

arm

Previous Approaches

- `clang/lib/Headers/opencl-c.h`
  - 17k lines, 800 KB, takes time to parse.
- Precompiled Headers (PCH).
  - Several MB large.
  - Declarations need to be hidden depending on preprocessor defines.
  - Not trivial to support different target devices / OpenCL versions simultaneously.
- Custom handling in `SemaChecking.cpp` / `CGBuiltin.cpp`
  - Modifying Clang is harder than modifying a list of declarations.
  - Still requires mapping to builtins (1000s of macros?).

TableGen Builtins

Compact TableGen description, new TableGen backend, and then:

- Predefine all OpenCL builtins at startup?

TableGen Builtins

Compact TableGen description, new TableGen backend, and then:

- ~~Predefine all OpenCL builtins at startup?~~
- Add all OpenCL builtins when first lookup fails?

TableGen Builtins

Compact TableGen description, new TableGen backend, and then:

- ~~Predefine all OpenCL builtins at startup?~~
- ~~Add all OpenCL builtins when first lookup fails?~~
- Add all OpenCL builtins with failed name when first lookup fails?
  - Minimizes impact on Clang startup time.
  - Reduces clutter in symbol table.

TableGen Builtin Specification

```
def Float : Type<"float", QualType<"FloatTy">>;
let Extension = Fp64TypeExt in {
  def Double : Type<"double", QualType<"DoubleTy">>;
}
let Extension = Fp16TypeExt in {
  def Half : Type<"half", QualType<"HalfTy">>;
}
```

TableGen Builtin Specification

```
def Float : Type<"float", QualType<"FloatTy">>;
let Extension = Fp64TypeExt in {
  def Double : Type<"double", QualType<"DoubleTy">>;
}
let Extension = Fp16TypeExt in {
  def Half : Type<"half", QualType<"HalfTy">>;
}
def AllFloats : TypeList<[Float, Double, Half]>;
def VecAndScalar : IntList<[1, 2, 3, 4, 8, 16]>;
def FGenTypeN : GenericType<AllFloats, VecAndScalar>;
```


arm

TableGen Builtin Specification

```
def Float : Type<"float", QualType<"FloatTy">>;
let Extension = Fp64TypeExt in {
  def Double : Type<"double", QualType<"DoubleTy">>;
}
let Extension = Fp16TypeExt in {
  def Half : Type<"half", QualType<"HalfTy">>;
}

def AllFloats : TypeList<[Float, Double, Half]>;
def VecAndScalar : IntList<[1, 2, 3, 4, 8, 16]>;
def FGenTypeN : GenericType<AllFloats, VecAndScalar>;

foreach name = ["cos", "cosh", "cospi"] in {
  def : Builtin<name, [FGenTypeN, FGenTypeN]>;
}
```

Autogenerated StringMatcher

Recognize a builtin name and return table entries describing that builtin.

```
switch (Name.size()) {
default: break;
case 3:  // 19 strings to match.
  switch (Name[0]) {
  case 'c':  // 3 strings to match.
    switch (Name[1]) {
    default: break;
    case 'o':  // 1 string to match.
      if (Name[2] != 's') break;
      return std::make_pair(731, 1);   // "cos"
...
case 4:  // 25 strings to match.
...
```


arm

TableGen generates... Tables!

```
...
// 730: get_global_linear_id, get_local_linear_id,
{ 1407, 1, 0, 0, 0, OCLE_null, 200, 0 },
// 731: acos, acosh, acospi, asin, asinh, asinpi, cos, cosh, cospi, ...
{ 1408, 2, 0, 1, 0, OCLE_null, 100, 0 },
// 732: nan
...
```


arm

TableGen generates... Tables!

```
...
// 730: get_global_linear_id, get_local_linear_id,
{ 1407, 1, 0, 0, 0, OCLE_null, 200, 0 },
// 731: acos, acosh, acospi, asin, asinh, asinpi, cos, cosh, cospi, ...
{ 1408, 2, 0, 1, 0, OCLE_null, 100, 0 },
// 732: nan
...
// Signature 1407
62,
// Signature 1408
14, 14,
...
```


arm

TableGen generates... Tables!

```
...
// 730: get_global_linear_id, get_local_linear_id,
{ 1407, 1, 0, 0, 0, OCLE_null, 200, 0 },
// 731: acos, acosh, acospi, asin, asinh, asinpi, cos, cosh, cospi, ...
{ 1408, 2, 0, 1, 0, OCLE_null, 100, 0 },
// 732: nan
...
// Signature 1407
62,
// Signature 1408
14, 14,
...
// Type 14
{OCLT_FGenTypeN, 0, 0, 0, 0, OCLAQ_None, clang::LangAS::Default},
// Type 62
{OCLT_size_t, 1, 0, 0, 0, OCLAQ_None, clang::LangAS::Default},
```


arm

Status and Future Work

- Patches are currently being reviewed / committed.
- Testing for completeness and correctness is an open challenge.
- Extend to non-OpenCL Clang builtins (Builtins.def).

Thanks