# Apex Builtins

This document was generated on 13-August-2018.

Build #530, based on clang 4.0.1

---

This is the list of low level APEX builtins. Each builtins map to one asm instruction. These are also used by the user level instrinsics defined in "`include/ewl2/apex/intrinsics.hpp`".

For details on the user level instrinsics, please refer to the document `ApexIntrinsics.pdf` For details on the ISA, please refer to the document `HDD_10294_14_08_APEX2_Inst_Ref_Guide.pdf`

The examples in this document are functions that contain only the builtin as the body. Because of register constraints, transfers may appear in order to enforce the calling convention, but the builtin will map to only 1 instruction.

For builtins that have a "parameter" marked with `const`, the examples are provided for values of `0` and `1`. When using these builtins with `const` parameters, the value given should be a compile time constant, not a variable. When the parameter is not marked with `const`, the value can be either a variable or a constant.

```
/* extract lower 16-bit from 32-bit vector */
vec16s __builtin_apex_vec32_get_lo(vec32s);
```

```
/* extract upper 16-bit from 32-bit vector */
vec16s __builtin_apex_vec32_get_hi(vec32s);
```

```
/* create 32-bit vector from two 16-bit fragment */
vec32s __builtin_apex_vec32_pack(vec16s lo, vec16s hi);
```

```
int __builtin_apex_sext(int a, int b) {
   xtd r1, r2, r3
}
```

```
int __builtin_apex_clb(int a) {
   clb r1, r2
}
```

```
int __builtin_apex_haddu(int a, int b) {
   hadduu r1, r2, r3
}
```

```
int __builtin_apex_hadds(int a, int b) {
   haddss r1, r2, r3
}
```

```
int __builtin_apex_rhaddu(int a, int b) {
   rhadduu r1, r2, r3
}
```

```
int __builtin_apex_rhadds(int a, int b) {
   rhaddss r1, r2, r3
}
```

```
int __builtin_apex_vget(vec16s a, int i) {
   vex r1, v0, r2
}
```

```
vec16s __builtin_apex_vput(vec16s s0, int s1, int i) {
   vwe v0, v1, r2, r1
}
```

```
vec16s __builtin_apex_vputv(vec16s s0, vec16s s1, int i) {
   vwe v0, v1, r1, v2
}
```

```
vec16s __builtin_apex_vputvv(vec16s s0, vec16s s1, vec16s s2) {
   vwe v0, v1, v3, v2
}
```

```
int __builtin_apex_vget_vc(vbool s0, const unsigned sel) {
 if (sel == 0) {
   vex_vc r1, vc0, #0
 }
 else if (sel == 1) {
   vex_vc r1, vc0, #1
 }
 else if ...
}
```

```
vbool __builtin_apex_vput_vc(int s0, const unsigned sel) {
 if (sel == 0) {
   vput vc0, r1, #0
 }
 else if (sel == 1) {
   vput vc0, r1, #1
 }
 else if ...
}
```

```
vec16s __builtin_apex_vaddr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vadd v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vadd v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vsubr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vsub v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vsub v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vandr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vand v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vand v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vxorr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vxor v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vxor v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vorr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vor v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vor v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vsext(vec16s a, vec16s b) {
   vxtd v0, v1, v2
}
```

```
vec16s __builtin_apex_vsexti(vec16s a, const unsigned b) {
 if (b == 0) {
   vxtd v0, v1, #0
 }
 else if (b == 1) {
   vxtd v0, v1, #1
 }
 else if ...
```

```
}
```

```
vec16s __builtin_apex_vsextr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vxtd v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vxtd v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16u __builtin_apex_vhaddu(vec16u a, vec16u b) {
   vhadduu v0, v1, v2
}
```

```
vec16s __builtin_apex_vhadds(vec16s a, vec16s b) {
   vhaddss v0, v1, v2
}
```

```
vec16u __builtin_apex_vrhaddu(vec16u a, vec16u b) {
   vrhadduu v0, v1, v2
}
```

```
vec16s __builtin_apex_vrhadds(vec16s a, vec16s b) {
   vrhaddss v0, v1, v2
}
```

```
vec16u __builtin_apex_vhaddur(vec16u a, unsigned b, const unsigned e) {
 if (e == 0) {
   vhadduu v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vhadduu v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vhaddsr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vhaddss v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vhaddss v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16u __builtin_apex_vrhaddur(vec16u a, unsigned b, const unsigned e) {
 if (e == 0) {
```

```
   vrhadduu v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vrhadduu v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vrhaddsr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vrhaddss v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vrhaddss v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vabs_diff(vec16s a, vec16s b) {
   vabs_diff v0, v1, v2
}
```

```
vec16u __builtin_apex_vabs_diffu(vec16u a, vec16u b) {
   vabs_diffu v0, v1, v2
}
```

```
vec16u __builtin_apex_vabs(vec16s a) {
   vabs v0, v1
}
```

```
vec16s __builtin_apex_vabs_diffr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vabs_diff v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vabs_diff v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16u __builtin_apex_vabs_diffur(vec16u a, int b, const unsigned e) {
 if (e == 0) {
   vabs_diffu v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vabs_diffu v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vadd_sat(vec16s a, vec16s b) {
   vadd_sat v0, v1, v2
}
```

```
vec16u __builtin_apex_vadd_satu(vec16u a, vec16u b) {
   vadd_satu v0, v1, v2
}
```

```
vec16s __builtin_apex_vsub_sat(vec16s a, vec16s b) {
   vsub_sat v0, v1, v2
}
```

```
vec16u __builtin_apex_vsub_satu(vec16u a, vec16u b) {
   vsub_satu v0, v1, v2
}
```

```
vec16s __builtin_apex_vadd_satr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vadd_sat v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vadd_sat v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vsub_satr(vec16s a, int b, const unsigned e) {
 if (e == 0) {
   vsub_sat v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vsub_sat v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16u __builtin_apex_vadd_satur(vec16u a, unsigned b, const unsigned e) {
 if (e == 0) {
   vadd_satu v0, v1, r1 [15:0]
 }
 else if (e == 1) {
   vadd_satu v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16u __builtin_apex_vsub_satur(vec16u a, unsigned b, const unsigned e) {
 if (e == 0) {
   vsub_satu v0, v1, r1 [15:0]
 }
```

```
 else if (e == 1) {
   vsub_satu v0, v1, r1 [31:16]
 }
 else if ...
}
```

```
vec16s __builtin_apex_vsat_vv(vec16s v, vec16s l, vec16s u) {
   vsat v0, v1, v2, v3
}
```

```
vec16s __builtin_apex_vsat_vr(vec16s v, vec16s l, int u) {
   vsat v0, v1, v2, r1
}
```

```
vec16s __builtin_apex_vsat_rv(vec16s v, int l, vec16s u) {
   vsat v0, v1, r1, v2
}
```

```
vec16s __builtin_apex_vsat_rr(vec16s v, int l, int u) {
   vsat v0, v1, r1, r2
}
```

```
vec16s __builtin_apex_vclz(vec16s a) {
   vclz v0, v1
}
```

```
vec16s __builtin_apex_vcld(vec16u a) {
   vcld v0, v1
}
```

```
vec16s __builtin_apex_vpcnt(vec16s a) {
   vpcnt v0, v1
}
```

```
vec16s __builtin_apex_vasb(vec16s a, vec16s b, vbool c, const unsigned f) {
 if (f == 0) {
   vasb v0, v1, v2, vc0, #0
 }
 else if (f == 1) {
   vasb v0, v1, v2, vc0, #1
 }
 else if ...
}
```

```
vec16s __builtin_apex_vasbr0(vec16s a, int b, vbool c, const unsigned f) {
 if (f == 0) {
   vasb v0, v1, r1, vc0, #0
 }
 else if (f == 1) {
```

```
   vasb v0, v1, r1, vc0, #1
 }
 else if ...
}
```

```
vec16s __builtin_apex_vasbr1(vec16s a, int b, vbool c, const unsigned f) {
 if (f == 0) {
   vasb v0, v1, r1, vc0, #0
 }
 else if (f == 1) {
   vasb v0, v1, r1, vc0, #1
 }
 else if ...
}
```

```
vec16s __builtin_apex_vasbs(vec16s a, vec16s b) {
   vasbs v0, v1, v2
}
```

```
vec16s __builtin_apex_vasbsvr(vec16s a, int b) {
   vasbs v0, v1, r1
}
```

```
vec16s __builtin_apex_vasbsrv(vec16s a, int b) {
   vasbs v0, v1, r1
}
```

```
vec16s __builtin_apex_vilb(vec16s a) {
   vilb v0, v1
}
```

```
vec16u __builtin_apex_vilbu(vec16s a) {
   vilbu v0, v1
}
```

```
vec16s __builtin_apex_vilw(vec16s a) {
   vilw v0, v1
}
```

```
void __builtin_apex_visb(vec16s a, vec16s b) {
   visb v0, v1
}
```

```
void __builtin_apex_visw(vec16s a, vec16s b) {
   visw v0, v1
}
```

```
vec16s __builtin_apex_vmul_lulu(vec16s a, vec16s b) {
   vmul_lulu v0, v1, v2
}
```

```
vec16s __builtin_apex_vmul_lslu(vec16s a, vec16s b) {
   vmul_lslu v0, v1, v2
}
```

```
vec16s __builtin_apex_vmul_lsls(vec16s a, vec16s b) {
   vmul_lsls v0, v1, v2
}
```

```
vec16s __builtin_apex_vmul_hulu(vec16s a, vec16s b) {
   vmul_hulu v0, v1, v2
}
```

```
vec16s __builtin_apex_vmul_huls(vec16s a, vec16s b) {
   vmul_huls v0, v1, v2
}
```

```
vec16s __builtin_apex_vmul_hslu(vec16s a, vec16s b) {
   vmul_hslu v0, v1, v2
}
```

```
vec16s __builtin_apex_vmul_hsls(vec16s a, vec16s b) {
   vmul_hsls v0, v1, v2
}
```

```
vec16s __builtin_apex_vmul_huhu(vec16s a, vec16s b) {
   vmul_huhu v0, v1, v2
}
```

```
vec16s __builtin_apex_vmul_hshu(vec16s a, vec16s b) {
   vmul_hshu v0, v1, v2
}
```

```
vec16s __builtin_apex_vmul_hshs(vec16s a, vec16s b) {
   vmul_hshs v0, v1, v2
}
```

```
vec16s __builtin_apex_vrl(vec16s a, vec16s b) {
   vrl v0, v1, v2
}
```

```
vec16s __builtin_apex_vslc(vbool c, vec16s a) {
   vsll_vc v0, vc0, v1
}
```

```
vec16s __builtin_apex_vsrc(vbool c, vec16s a) {
   vsrl_vc v0, vc0, v1
}
```

```
vbool __builtin_apex_vcsr(vbool a, vbool b) {
   vcsr vc0, vc1, vc2
}
```

```
vbool __builtin_apex_vcsl(vbool a, vbool b) {
   vcsl vc0, vc1, vc2
}
```

```
int __builtin_apex_vany(vbool a) {
   vany r1, vc0
}
```

```
int __builtin_apex_vall(vbool a) {
   vall r1, vc0
}
```

```
int __builtin_apex_padd(int a, const unsigned b) {
 if (b == 0) {
   padd r2, #0
   or r1, r2, r2
 }
 else if (b == 1) {
   padd r2, #1
   or r1, r2, r2
 }
 else if ...
}
```

```
int __builtin_apex_paddr(int a, const unsigned b) {
 if (b == 0) {
   add r1, r2, #0
 }
 else if (b == 1) {
   add r1, r2, #1
 }
 else if ...
}
```

```
int __builtin_apex_paddrr(int a, int b) {
   add r1, r2, r3
}
```

```
int __builtin_apex_mov(int a) {
   mov r1, r2
}
```

```
void __builtin_apex_vcs_push(vbool a) {
   vcspush vc0
}
```

```
void __builtin_apex_vcs_flip() {
   vcsflip
}
```

```
void __builtin_apex_vcs_pop() {
   vcspop
}
```

```
vbool __builtin_apex_vcs_get() {
   vcmv vc0, vcsptr
}
```

```
vbool __builtin_apex_vcs_inv_get() {
   vcinv vc0, vcsptr
}
```

```
vec16s __builtin_apex_vmrlv(vec16s a, vec16s b) {
   vmrlv v0, v1, v2
}
```

```
vec16s __builtin_apex_vmrrv(vec16s a, vec16s b) {
   vmrrv v0, v1, v2
}
```

```
vec16s __builtin_apex_vmrlr(vec16s a, int b) {
   vmrlr v0, v1, r1
}
```

```
vec16s __builtin_apex_vmrrr(vec16s a, int b) {
   vmrrr v0, v1, r1
}
```

```
void __builtin_apex_wait(const unsigned a) {
 if (a == 0) {
   wait #0
 }
 else if (a == 1) {
   wait #1
 }
 else if ...
}
```

```
void __builtin_apex_swbreak() {
   swbrk
```

```
}
```