U++ - Bug #2210
se/FP80 crashes pdb debugger

08/06/2021 09:53 AM - Miroslav Fidler

Status:	New	**Start date:**	08/06/2021
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour

Description

```
#include <Core/Core.h>

using namespace Upp;

#ifdef SIZEOF_INT128    // GNU C
 static inline
 uint64_t mulhi64(uint64_t a, uint64_t b) {
    unsigned __int128 prod =  a * (unsigned __int128)b;
    return prod >> 64;
 }

#elif defined(_M_X64) || defined(_M_ARM64)    // MSVC
  // MSVC for x86-64 or AArch64
  // possibly also  || defined(_M_IA64) || defined(_WIN64)
  // but the docs only guarantee x86-64!  Don't use just _WIN64; it doesn't include AArch64 Android / Linux
// https://docs.microsoft.com/en-gb/cpp/intrinsics/umulh
  #include <intrin.h>
  #define mulhi64 __umulh


#elif defined(_M_IA64) // || defined(_M_ARM)      // MSVC again
  // https://docs.microsoft.com/en-gb/cpp/intrinsics/umul128
  // incorrectly say that _umul128 is available for ARM
  // which would be weird because there's no single insn on AArch32
  #include <intrin.h>
  static inline
  uint64_t mulhi64(uint64_t a, uint64_t b) {
    unsigned __int64 HighProduct;
    (void)_umul128(a, b, &HighProduct);
    return HighProduct;
  }

#else

   1. undef HAVE_FAST_mul64
        uint64_t mulhi64(uint64_t a, uint64_t b);  // non-inline prototype
        // or you might want to define @craigster0's version here so it can inline.
      #endif

struct FP80 {
   uint64 m;
   int    e;
```

```
    void Set(uint64 x) {
        int q = SignificantBits64(x);
        m = x << (64 - q);
        e = q - 64;
      }
    void operator*=(const FP80& b) {
        m = mulhi64(m, b.m);
        e += b.e + 64;
      }
    void Div2()
    {
        e--;
      }
    String ToString() const { return AsString(m) << ", exp: " << e << ", dbl: " << m * pow(2, e); }
    };

FP80 operator+(FP80 a, FP80 b){
  int shift = a.e - b.e; // <<< Crash here on breakpoint and going Locals
  if(shift < 0) {
    Swap(a, b);
    shift = -shift;
  }
  if(shift > 64)
    return a;
  a.m += b.m >> shift;
  return a;
}

CONSOLE_APP_MAIN{
  FP80 x;
  x.Set(1);
DDUMP;
  FP80 ten;
  ten.Set(10);
  x *= ten;
DDUMP;
DDUMP;
DDUMP;
}
```