Specification for features detect and platform configuration control library.

AVX_OS.DLL.

Version 2.02.

Library AVX_OS.DLL detects platform functional features: AVX (Advanced Vector Extensions) instructions support for vector data calculations, cache memory configuration and size for L1, L2, L3 caches. Results passed to high level program for select one of mathematics libraries, associated with functionality, supported by this platform and optimizing data block sizes for cache memory size and number of threads/cores. Library detects processor and operating system features. Note that for support AVX, required support this instruction set by processor and support 256-bit context save-restore by OS task switch procedure. CPU features detected by CPUID instruction. OS context management features detected by XCR0 system register bits.

1. Function DetectVXLib. Detect Vector eXtension Library. Detect supported vector instruction sets and select appropriate library.

- Returns structure, contains list of mathematics libraries and information about possibility this libraries usage on current detected platform.
- Possibility of libraries usage detected by central processor and operating system features.
- Address for built library information structure passed to this function from high level program, as input parameter.

Returned library information structure can declare maximum 20 libraries or 20 groups of libraries. Library suffix is unique for each group, but same in the group. Groups selected by platform features. Libraries in each group can be classified as unified (Common) or architecture-specific (Specific). Specific can use interface, designed for concrete processor architecture and extreme optimized for this architecture. Libraries classification inside group (division to Common and Specific), not relevant with this function.

Each group of libraries uses descriptor with 16 bytes size. That means full structure size is 20*16=320 bytes. Each descriptor contain 2 bytes-flags, indicates possibility of usage this libraries group for current detected processor and operating system. Also contain 10 byte group suffix and 32-bit value contain vector register length in bytes. Alignment factor in bytes calculated as this value divided by 8.

One group descriptor format:

```
T_VXLibRec = Packed RECORD
  SupportCPU: AnsiChar;
  SupportOS: AnsiChar;
  VerName: ARRAY[0..9] OF AnsiChar;
  VRlen: Uint32;
END;
```

Group descriptor fields:

SupportCPU - byte (ASCII char), indicator of availability for this library group, this indicator set by check of central processor features. Char '-' (ASCII-code 2Dh) means this library load is impossible. Char '+' (ASCII-code 2Bh) means library load is possible. If this descriptor not used (empty), this field contain char '-'.

SupportOS - byte (ASCII char), indicator of availability for this library group, this indicator set by check of operating system features. Char '-' (ASCII-code 2Dh) means this library load is impossible. Char '+' (ASCII-code 2Bh) means library load is possible. If this descriptor not used (empty), this field contain char '-'.

VerName - string of 10 ASCII chars, contain suffix used for built names of libraries, associated with this group. All 10 chars must be initialized, for formatting restrictions smaller string with null-termination cannot be used. Smaller string can be augmented by underline char '_' (ASCII-code 5Fh). For empty descriptors, contains '-' at fields SupportCPU and SupportOS, this field must contain 10 zero bytes.

VRlen - length of vector register in bits. That means, alignment factor in bytes calculated by divide this value by 8. For empty descriptors, contains '-' at fields SupportCPU and SupportOS, this field must contain zero byte.

Function declaration.

```
PROCEDURE  DetectVXLib  ( ProcessorAndOsSupport:  P_VXLibMatrix );
```

Input parameters.

1. Function accepts one parameter - pointer, and returns by this pointer 320 bytes structure. Structure contains 20 descriptors, format of this descriptors detailed above. Unlike calculation functions, pointer can be unaligned.

Output parameters.

320 bytes structure at address, passed by input pointer. Format of this structure described above.

Library suffixes at this list sorted by performance speed grow. Library useable if both associated indicator bytes SupportCPU and SupportOS contains char '+', that means support required CPU instructions sets by processor and operating system. High level program must select library with maximum level of performance speed by scan list from last descriptor to first and search descriptor with both bytes SupportCPU и SupportOS contains char '+'.

2. Function DetectCache. Detect size and configuration of cache memory.

- Returns array of 4 64-bit numbers, contains information about presence and size of cache memory (in bytes). This information for Data Cache and Unified Cache, instruction cache not reported by this function. Also returns number of execution threads per each core.

Assume that each CPU core has own cache memory levels L1 and L2. L1 cache is separated as data cache and instruction cache (from L1 function returns data cache only). L2 is unified for data and instructions. Logical processor, created by Hyper-Threading technology, share L1 and L2. This fact must be considered by high level program when select optimal sizes of data block. L3 cache is unified for data and instructions and shared by all cores. This means, only physically multiprocessing platform can has more than one block of L3.

Typical CPU has 1 thread per core if Hyper-Threading (HT) technology not used or 2 threads per core if HT used. But passing number of threads (not HT flag as one bit) allows support system with number of threads above 2.

Cache memory configuration descriptor format.

T_CacheInfo = RECORD

```
    L1data: Uint64;
    L2unified: Uint64;
    L3unified: Uint64;
    ThreadsCount: Uint64;
END;
```

Descriptor fields details:

L1data - size of L1 data cache (bytes) per one execution thread.

L2unified - size of L2 unified cache (bytes) per one execution thread.

L3unified - size of L3 unified cache (bytes) per physical processor.

ThreadsInfo - Number of execution threads (logical processors) per core.

Function declaration.

```
// Cache memory detect
FUNCTION DetectCache
        (CacheInfo  : P_Uint64; )  // Pointer to output block
     : T_Uint32;                   // Returned status
```

Input parameters.

1. Function accepts one parameter - pointer, and returns by this pointer 32 bytes structure. Structure contains 4 64-bit numbers, detailed above. Unlike calculation functions, pointer can be unaligned.

Output parameters.

1. Status. Zero value means function executed successfully and output block is valid. Non-zero value means error and output block is invalid.

2. Structure, size is 32 bytes, format described above. This structure located at address, passed by first parameter - pointer.