| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [LibVsb/include](http://docs.google.com/index.html) - vsb\_types.h (source / [functions](http://docs.google.com/vsb_types.h.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 3 | 3 | 100.0 % | | Date: | 2012-03-23 |  | Functions: | 0 | 0 | - | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 1 | 2 | 50.0 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : /\*--------------------------------------------------------------------------  2 : : \* Copyright (c) 2001, Toshiba Machine Co, Digital GlobalSoft Ltd.  3 : : \* All Rights Reserved.  4 : : \* These technical data are subject to export control laws of Japan  5 : : \* regulations, and diversion contrary thereto is prohibited.  6 : : \*   7 : : \* Module : VSB Routines  8 : : \* Design Document : VSB Library - High Level Design (Rev 0.76)  9 : : \* Author : santhosh.kudva@digital.com  10 : : \* Date : 3rd April, 2003  11 : : \*   12 : : \* File names prefixed with vsb\_ contain definitions used privately by the   13 : : \* library   14 : : \*   15 : : \* RCS Keywords  16 : : \* $Id: vsb\_types.h,v 1.3 2011/07/05 11:20:39 asarkar Exp $  17 : : \* $Log: vsb\_types.h,v $  18 : : \* Revision 1.3 2011/07/05 11:20:39 asarkar  19 : : \* \*\*\* empty log message \*\*\*  20 : : \*  21 : : \* Revision 1.2 2011/05/23 10:48:13 asarkar  22 : : \* \*\*\* empty log message \*\*\*  23 : : \*  24 : : \* Revision 1.1 2011/04/26 11:22:48 asarkar  25 : : \* VSB Library  26 : : \*  27 : : \* Revision 1.2 2007/08/07 15:53:19 debarshi  28 : : \* Fixed some bugs on array cell processing  29 : : \*  30 : : \* Revision 1.1 2007/07/05 06:47:41 santanu  31 : : \* Santanu::Second phase megre  32 : : \*  33 : : \* Revision 1.2 2007/02/05 11:46:12 santanu  34 : : \* EBM5K updated to version 1.67  35 : : \*  36 : : \* Revision 1.4 2005/02/26 20:52:56 kumar  37 : : \* Changes related to Logging/Monitoring  38 : : \*  39 : : \* Revision 1.3 2004/06/09 22:18:32 gopal  40 : : \* Changes related to Linux Porting  41 : : \*  42 : : \* Revision 1.2 2004/01/23 18:34:11 gopal  43 : : \* Copied new files from non-dump mode CVS  44 : : \*  45 : : \* Revision 1.5 2004/01/23 17:46:55 raj  46 : : \* Bug Di #189  47 : : \* 1) VSB Data types are not in sync with VSB-12 specification Rev 0.962  48 : : \* Data type changes done accroding to specification  49 : : \*  50 : : \* Review: Gopal  51 : : \*  52 : : \* Revision 1.4 2003/09/08 21:52:37 kudva  53 : : \* Bug ID #24 : Patterns have 5 AU Error (AR-Linear-Term, No Scale, AU Conversion Factor = 10)  54 : : \* Analysis : An Array Representation can have a non integral pitch. So to solve this,  55 : : \* 1. RoundINT was added to calculate pattern position (PP) from pitch.  56 : : \* 2. The PP was Translated. (This involves one more RoundINT)  57 : : \* Due to this the PP which was translated had a lower precision  58 : : \*  59 : : \* Eg. if pattern position was PP.X0 = 123.45 (after using pitch to calculate it)  60 : : \* RoundINT would result in PP.X0 = 123  61 : : \* AU Conversion = PP.X0 \* 10 = 1230  62 : : \*  63 : : \* However, if PP.X0 = 123.45  64 : : \* AU Conversion of this value would be 1234.5 and on round int32\_t. PP.X0 = 1235.  65 : : \* This is a 5 AU difference.  66 : : \*  67 : : \* Fix : To remove the RoundINT, the PP must be kept as a double. The same problem was also anticipated for CR. Therefore, vsb now implements a new type vsb\_ep and vsb\_pp, whose members are of primitive type double. EP & PP are now maintained in vsb\_ep and vsb\_pp structures till they have to be returned to the caller. This results in at most 0.5 AU of error from VSB for Pattern positions.  68 : : \*  69 : : \* Reviewed by - Gopal Mor  70 : : \* Checked in by - Santhosh Kudva  71 : : \*  72 : : \* Revision 1.3 2003/07/25 20:26:34 kudva  73 : : \* 2 Byte Alignment Changes  74 : : \*  75 : : \* Changes the Segment Header to reflect the new structure as proposed in  76 : : \* VSB-12 Rev. 0.962 (X0Y0WordLength is now 4 bits)  77 : : \*  78 : : \* Reviewed by - Shiva  79 : : \*  80 : : \* Revision 1.2 2003/04/29 21:36:53 narayan  81 : : \* Changes  82 : : \* 1. PH1 and PH2 had the Flag fields in wrong endian order, updated to reflect the correct endianess (reversed the order in the struct)  83 : : \*  84 : : \* Checked in - Santhosh Kudva  85 : : \* Reviewed By - Narayana Reddy  86 : : \*  87 : : \* Revision 1.1 2003/04/23 21:43:42 kudva  88 : : \* VSB First Code Checkin - First Baseline - Santhosh Kudva  89 : : \*  90 : : \* -------------------------------------------------------------------------\*/  91 : :   92 : : #ifndef \_VSB\_TYPES\_H\_  93 : : #define \_VSB\_TYPES\_H\_  94 : :   95 : : #include "Porting.h"  96 : :   97 : : #ifdef \_\_cplusplus  98 : : extern "C" {  99 : : #endif   100 : :   101 : : namespace nft {  102 : : typedef enum \_orientation  103 : : {   104 : : VERTICAL = 0,  105 : : HORIZONTAL = 1  106 : : } eOrientation\_t;  107 : :   108 : :   109 : : typedef enum \_RefSegments   110 : : {  111 : : VSB\_REF\_HEADER = 0,   112 : : VSB\_REF\_BLOCK\_POINTER = 1,   113 : : VSB\_REF\_EXTENSION = 2,   114 : : VSB\_REF\_BLOCK\_DEF = 3  115 : : } RefSegments ;  116 : :   117 : : typedef enum \_PatSegments   118 : : {  119 : : VSB\_PAT\_HEADER = 0,  120 : : VSB\_PAT\_SEG\_POINTER = 1,  121 : : VSB\_PAT\_EXTENSION = 2,  122 : : VSB\_PAT\_DATA\_COMPOUND = 3   123 : : } PatSegments ;   124 : :   125 : : typedef enum \_LinkSegments   126 : : {  127 : : VSB\_LINK\_HEADER = 0,  128 : : VSB\_LINK\_REL\_TBL = 1,  129 : : VSB\_LINK\_CELL\_OP\_TBL = 2,  130 : : VSB\_LINK\_FILE\_MAP\_TBL = 3  131 : : } LinkSegments ;  132 : :   133 : :   134 : :   135 : :   136 : :   137 : : typedef enum \_vsbCellType   138 : : {   139 : : SINGLE\_CELL = 0,   140 : : ARRAY\_CELL = 1   141 : : } vsbCellType ;  142 : :   143 : : typedef struct \_BlkMap  144 : : {  145 : : int32\_t IDx ;  146 : : int32\_t IDy ;  147 : : int32\_t CellId ;  148 : : } BlkMap ;  149 : :   150 : : typedef struct \_vsbBlockPointer  151 : : {  152 : : int32\_t BlockPointer ;  153 : : int8\_t IDx ;  154 : : int8\_t IDy ;  155 : : } vsbBlockPointer ;  156 : :   157 : : typedef struct \_vsbBlockDefinitionHeader  158 : : {  159 : : uint32\_t BlockDataLength ;  160 : : int8\_t IDx ;  161 : : int8\_t IDy ;  162 : : uint32\_t NumCellLocations ;  163 : : } vsbBlockDefinitionHeader ;  164 : :   165 : : typedef struct \_vsbCompCellPatternDataHeader //Compression Support  166 : : {  167 : : unsigned int compLength;  168 : : unsigned int unCompLength;  169 : : } vsbCompCellPatternDataHeader ;  170 : :   171 : : typedef struct \_MagicNumber  172 : : {  173 : : unsigned int code\_c : 8 ;  174 : : unsigned int code\_f : 8 ;  175 : : unsigned int flag : 8 ;  176 : : unsigned int code\_e : 8 ;  177 : : } MagicNumber ;  178 : :   179 : : typedef struct \_vsbSingleCell  180 : : {  181 : : uint32\_t Dx : 29 ;  182 : : int32\_t Flag1 : 2 ;  183 : : int32\_t CellType : 1 ;  184 : : uint32\_t Dy : 29 ;  185 : : int32\_t Flag2 : 2 ;  186 : : int32\_t CellRefType : 1 ;  187 : : int32\_t Xb ;  188 : : int32\_t Yb ;  189 : : int32\_t RelationRecordIndex ;   190 : : } vsbSingleCell ;  191 : :   192 : : typedef struct \_vsbArrayCell  193 : : {  194 : : unsigned int Dx : 29 ;  195 : : int Flag1 : 2 ;  196 : : int CellType : 1 ;  197 : : unsigned int Dy : 29 ;  198 : : int Flag2 : 2 ;  199 : : int CellRefType : 1 ;  200 : : int Xb ;  201 : : int Yb ;  202 : : int Xe ;  203 : : int Ye ;  204 : : unsigned short int Nrepy ;  205 : : unsigned short int Nrepx ;  206 : : int RelationRecordIndex ;  207 : : } vsbArrayCell ;  208 : :   209 : : typedef struct \_vsbCellPatternDataHeader  210 : : {  211 : : uint32\_t CellPatternDataLength ;  212 : : uint32\_t Dx : 29 ;  213 : : uint32\_t flag1 : 3 ;  214 : : uint32\_t Dy : 29 ;  215 : : uint32\_t flag2 : 3 ;  216 : :   217 : : } vsbCellPatternDataHeader ;  218 : :   219 : : typedef struct \_vsbPatternSegmentHeader  220 : : {  221 : : unsigned int HeaderLength ;  222 : : unsigned int AU ;  223 : : unsigned int Reserved1:16 ;  224 : : unsigned int compressionType:3 ; //Compression Support  225 : : unsigned int Reserved2:5 ;  226 : : unsigned int X0Y0WordLength:4 ;  227 : : unsigned int isAligned:4 ;  228 : : unsigned int SegmentDataLength ;  229 : : unsigned int NumCellPatternData ;  230 : : unsigned int PatternDataType ;  231 : : } vsbPatternSegmentHeader ;  232 : :   233 : : /\*typedef struct \_vsbPatternSegmentHeader  234 : : {  235 : : uint32\_t HeaderLength ;  236 : : uint32\_t AU ;  237 : : uint32\_t Reserved:24 ;  238 : : uint32\_t X0Y0WordLength:4 ;  239 : : uint32\_t isAligned:4 ;  240 : : uint32\_t SegmentDataLength ;  241 : : uint32\_t NumCellPatternData ;  242 : : uint32\_t PatternDataType ;  243 : : } vsbPatternSegmentHeader ;  244 : : \*/  245 : :   246 : : typedef struct \_vsbPH1  247 : : {  248 : :   249 : : uint32\_t NumEP : 16;  250 : : uint32\_t isOptionAdded:1;  251 : : uint32\_t L1WordLength:2;  252 : : uint32\_t L2WordLength:2;  253 : : uint32\_t isOR2:1;  254 : : uint32\_t PPDirection:2;  255 : : uint32\_t code1: 8;   256 : :   257 : : } vsbPH1 ;  258 : :   259 : : typedef struct \_vsbPH2  260 : : {  261 : : uint32\_t isOptionAdded:1 ;  262 : : uint32\_t L1WordLength:2;  263 : : uint32\_t L2WordLength:2;  264 : : uint32\_t isOR2:1 ;  265 : : uint32\_t PPDirection:2 ;  266 : : uint32\_t code1 :8 ;  267 : : uint32\_t codea :8 ;  268 : : uint32\_t codes :8 ;   269 : : } vsbPH2 ;  270 : :   271 : : typedef struct \_vsbPH3  272 : : {  273 : : uint32\_t Nv :16 ;  274 : : uint32\_t codea :8 ;  275 : : uint32\_t codes :8;   276 : : }vsbPH3 ;  277 : :   278 : : //PH4 Handling  279 : : typedef struct \_vsbPH4\_1  280 : : {  281 : : unsigned int AI1 : 16;  282 : : unsigned int flag1 : 8;  283 : : unsigned int b : 8;  284 : : } vsbPH4\_1;  285 : :   286 : :   287 : : typedef struct \_vsbPH4\_2  288 : : {  289 : : unsigned int AI2 : 24;  290 : : unsigned int b : 8;  291 : : } vsbPH4\_2;  292 : :   293 : :   294 : : //New code added by Interra ===>  295 : : inline double getAUFactor(vsbPatternSegmentHeader \*SegmentHeader)  296 : 2522 : {  297 [ + - ]: 1261 : return BE2LE\_32(SegmentHeader->AU) ;  298 : : }  299 : : inline int getX0Y0WordLength(vsbPatternSegmentHeader \*SegmentHeader)  300 : 1261 : {  301 : : return SegmentHeader->X0Y0WordLength ;  302 : : }  303 : : //<== new copde ends here  304 : :   305 : :   306 : :   307 : : typedef struct \_vsbEP  308 : : {  309 : : unsigned char PatCode ;  310 : : double X0 ;  311 : : double Y0 ;  312 : : double L1 ;  313 : : double L2 ;  314 : : double DLx ;  315 : : double DLy ;  316 : : } vsbEP ;  317 : :   318 : : typedef struct \_vsbPP  319 : : {  320 : : double X0 ;  321 : : double Y0 ;  322 : : } vsbPP ;  323 : :   324 : :   325 : : } //namespace nft  326 : : #ifdef \_\_cplusplus  327 : : }  328 : : #endif   329 : :   330 : : #endif /\* \_VSB\_TYPES\_H\_ \*/ |

|  |
| --- |
| Generated by: [LCOV version 1.9](http://ltp.sourceforge.net/coverage/lcov.php) |