| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [LibVsb/src](http://docs.google.com/index.html) - vsb\_patternNR.cpp (source / [functions](http://docs.google.com/vsb_patternNR.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 69 | 73 | 94.5 % | | Date: | 2012-03-23 |  | Functions: | 2 | 2 | 100.0 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 14 | 20 | 70.0 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : /\*--------------------------------------------------------------------------  2 : : \* Copyright (c) 2003, NuFlare Technology Inc., 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 Normal Representation Extraction Routines  8 : : \* Design Document : VSB Library - High Level Design (Rev 0.76)  9 : : \* Author : santhosh.kudva@digital.com  10 : : \* Date : 14th April, 2003  11 : : \*   12 : : \* File names prefixed with vsb\_ contain routines used privately by the library   13 : : \*   14 : : \* RCS Keywords  15 : : \* $Id: vsb\_patternNR.cpp,v 1.8 2012/02/10 05:07:20 santanu Exp $  16 : : \* $Log: vsb\_patternNR.cpp,v $  17 : : \* Revision 1.8 2012/02/10 05:07:20 santanu  18 : : \* Fix for E9K-80  19 : : \*  20 : : \* Revision 1.7 2011/11/02 06:34:34 eupa  21 : : \* Code for healthcheck, dumping data for errors and its testing, check for positive dpb no  22 : : \*  23 : : \* Revision 1.6 2011/09/01 09:49:16 abua  24 : : \* SD related changes  25 : : \*  26 : : \* Revision 1.5 2011/08/18 11:07:00 abua  27 : : \* initial commit for SelfDiagnostics  28 : : \*  29 : : \* Revision 1.4 2011/07/05 11:20:39 asarkar  30 : : \* \*\*\* empty log message \*\*\*  31 : : \*  32 : : \* Revision 1.3 2011/05/23 10:47:40 asarkar  33 : : \* \*\*\* empty log message \*\*\*  34 : : \*  35 : : \* Revision 1.2 2011/05/19 09:23:45 asarkar  36 : : \* Updated  37 : : \*  38 : : \* Revision 1.1 2011/04/26 11:22:49 asarkar  39 : : \* VSB Library  40 : : \*  41 : : \* Revision 1.1 2007/07/05 06:47:42 santanu  42 : : \* Santanu::Second phase megre  43 : : \*  44 : : \* Revision 1.2 2007/02/05 11:46:12 santanu  45 : : \* EBM5K updated to version 1.67  46 : : \*  47 : : \* Revision 1.5 2005/03/15 00:30:15 ram  48 : : \* Added the changes for Self-Diagnostics macro. The SELF\_DIAG is changed to SELFDIAG\_PMD as a part of pattern shift detection.  49 : : \*  50 : : \* Revision 1.4 2004/09/20 21:57:29 ram  51 : : \* Changes related to Self Diagnostics  52 : : \*  53 : : \* Revision 1.3 2003/12/16 21:42:49 gopal  54 : : \* 1. Merged from main CVS after performance related changes  55 : : \*  56 : : \* Revision 1.7 2003/12/11 16:41:33 gopal  57 : : \* 1. Merged performance related changes from exp kit 6-b  58 : : \*  59 : : \* Revision 1.6 2003/11/10 15:58:32 raj  60 : : \* CR PG Keep bug fixes  61 : : \* 1. Termination conditions  62 : : \* 2. Flag resetting  63 : : \* 3. coordinate calculations  64 : : \* 4. counter updates  65 : : \*  66 : : \* Revision 1.5 2003/11/10 14:48:53 raj  67 : : \* CR PG Keep changes done by Santhosh Kudva San after CR PG Keep design  68 : : \*  69 : : \* 1. Complete changes for CR functions.  70 : : \* 2. related changes in fetching OR , NR and AR  71 : : \* 3. translate function changes  72 : : \* 4. Context data structure changes  73 : : \* ( this code release date 10 Oct 2003)  74 : : \* Revision 1.4 2003/09/08 21:52:37 kudva  75 : : \* Bug ID #24 : Patterns have 5 AU Error (AR-Linear-Term, No Scale, AU Conversion Factor = 10)  76 : : \* Analysis : An Array Representation can have a non integral pitch. So to solve this,  77 : : \* 1. RoundINT was added to calculate pattern position (PP) from pitch.  78 : : \* 2. The PP was Translated. (This involves one more RoundINT)  79 : : \* Due to this the PP which was translated had a lower precision  80 : : \*  81 : : \* Eg. if pattern position was PP.X0 = 123.45 (after using pitch to calculate it)  82 : : \* RoundINT would result in PP.X0 = 123  83 : : \* AU Conversion = PP.X0 \* 10 = 1230  84 : : \*  85 : : \* However, if PP.X0 = 123.45  86 : : \* AU Conversion of this value would be 1234.5 and on round int. PP.X0 = 1235.  87 : : \* This is a 5 AU difference.  88 : : \*  89 : : \* 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.  90 : : \*  91 : : \* Reviewed by - Gopal Mor  92 : : \* Checked in by - Santhosh Kudva  93 : : \*  94 : : \* Revision 1.3 2003/04/29 22:29:59 narayan  95 : : \* Changes  96 : : \* 1. L1, L2 byte length defined in Ph1 and Ph2 flags start from 0, therefore 1 needs to  97 : : \* be added  98 : : \* 2. Initialization of CurrEPNum in NRContext  99 : : \*  100 : : \* Checked in by - Santhosh Kudva  101 : : \* Reviewed by - Narayana Reddy  102 : : \*  103 : : \* Revision 1.2 2003/04/25 17:28:59 narayan  104 : : \* Updated private declerations of function prototypes with extern "C" to  105 : : \* prevent Name Mangling  106 : : \*  107 : : \* Reviewed By - Narayana Reddy  108 : : \* Checked in By - Santhosh Kudva  109 : : \*  110 : : \* Revision 1.1 2003/04/23 21:43:42 kudva  111 : : \* VSB First Code Checkin - First Baseline - Santhosh Kudva  112 : : \*  113 : : \* -------------------------------------------------------------------------\*/  114 : :   115 : : #include <stdio.h>  116 : : #include "dec\_types.h"  117 : : #include "vsb\_types.h"  118 : : #include "vsb\_utils.h"  119 : : #include "vsb\_translate.h"  120 : : #include "context.h"  121 : : #include "PH4Flag.h"  122 : : #include "SDDB.h"  123 : : #include "HealthCheckCounter.h"  124 : : namespace nft {  125 : : /\* Private Decelerations \*/  126 : :   127 : : extern "C" int NRGetEP(GroupContext \*pgc, EP \*ep, int \*Nx, int \*Ny, int \*Xe, int \*Ye) ;  128 : :   129 : :   130 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  131 : : \* Name: NRGetEP  132 : : \* Functionality: Extracts one EP from a Normal Representation (NR : PH1 EPi+)  133 : : \* Input: Pointers to GroupContext, EP, Number of X Repetitions, Number of Y   134 : : \* Repetitions, Array end position X direction, Array end position Y direction  135 : : \* Output: Returns an EP as a Array Representation, Nx=Ny=1  136 : : \* Return value: Success or failure of the operation  137 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  138 : :   139 : : int NRGetEP(GroupContext \*pgc, EP \*ep, int \*Nx, int \*Ny, int \*Xe, int \*Ye)  140 : 5985 : {  141 : 5985 : HealthCheckCounter::instance()->incrCounter(  142 : : HealthCheckCounter::eUpdateFrequent);  143 : :   144 : 5985 : NRContext \*nrctx ;  145 : 5985 : vsbPH1 ph1 ;   146 : 5985 : int SkipSize ;  147 : 5985 : alias\_PH4\_1 alias\_PH4\_1Val;  148 : 5985 : alias\_PH4\_2 alias\_PH4\_2Val;  149 : 5985 : alias\_PH1 alias\_PH1Val;  150 : :   151 : :   152 : 5985 : nrctx = &(pgc->ppc.nrctx) ;  153 : :   154 [ + + ]: 5985 : if(nrctx->isEmpty)  155 : : {  156 : : //ph1 = (vsbPH1 \*)pgc->current ;  157 : 1577 : uint temp ;  158 : : //temp = vsbGetInt(pgc->current) ;  159 : 1577 : alias\_PH4\_1Val.value = alias\_PH4\_2Val.value = vsbGetInt(pgc->current) ;  160 : : //PH4 handling  161 : 1577 : if(1 == PH4Flag::instance()->getValue())  162 : : {  163 : 32 : vsbPH4\_1 ph4;  164 : : //ph4 =\*(vsbPH4\_0 \*)(&temp) ;  165 : 32 : ph4 = alias\_PH4\_1Val.ph4 ;  166 [ + + ][ - + ]: 32 : switch(ph4.b)  167 : : {  168 : : case 0xb1:  169 : : {  170 : 16 : vsbPH4\_1 ph4\_1;  171 : : //ph4\_1= \*(vsbPH4\_1 \*)(&temp);  172 : 16 : ph4\_1 = alias\_PH4\_1Val.ph4 ;  173 : 16 : pgc->attributeInfo=ph4\_1.AI1;  174 : 16 : pgc->current = pgc->current + sizeof(vsbPH4\_1);  175 : 16 : break;  176 : : }  177 : : case 0xb2:  178 : : {  179 : 16 : vsbPH4\_2 ph4\_2;  180 : : //ph4\_2 = \*(vsbPH4\_2 \*)(&temp);  181 : 16 : ph4\_2 = alias\_PH4\_2Val.ph4 ;  182 : 16 : pgc->attributeInfo = ph4\_2.AI2;  183 : 16 : pgc->current = pgc->current + sizeof(vsbPH4\_2);  184 : 16 : break;  185 : : }  186 : : }  187 : : }  188 : : else  189 : : {  190 : 1545 : pgc->attributeInfo=0;  191 : : }  192 : :   193 : : //temp = vsbGetInt(pgc->current) ;  194 : 1577 : alias\_PH1Val.value = vsbGetInt(pgc->current) ;  195 : : //ph1 = \*(vsbPH1 \*)(&temp) ;  196 : 1577 : ph1 = alias\_PH1Val.ph1;  197 : 1577 : nrctx->PatCode = ph1.code1 ;  198 : 1577 : try{  199 : 1577 : checkPatternCode(nrctx->PatCode);  200 : : }  201 [ - + ]: 32 : catch(DPL::Error \*e)  202 : : {  203 : 32 : char msg[1024];  204 : 32 : sprintf(msg,"Invalid Pattern code 0x%x\n",nrctx->PatCode);  205 : :   206 : 32 : ErrorHandler \*pErr = ErrorHandler::instance();  207 : 32 : pErr->errorAdd(NULL, (char\*)\_\_FILE\_\_, (char\*)\_\_FUNCTION\_\_, (int)\_\_LINE\_\_  208 : : ,ErrorHandler::eErrFunction,pErr->getErrorMsg(ErrorHandler::eErrFunction  209 : : ,"Invalid Pattern Code Error while reading NR pattern"));  210 : : }  211 : 1545 : nrctx->NumEP = ph1.NumEP ;  212 : 1545 : nrctx->L1WordLength = ph1.L1WordLength + 1 ;  213 : 1545 : nrctx->L2WordLength = ph1.L2WordLength + 1 ;  214 : 1545 : nrctx->NumFieldsInEP = vsbNumEPFields(nrctx->PatCode) ;  215 : 1545 : nrctx->CurrEPNum = 0 ;  216 : 1545 : nrctx->isEmpty = false ;  217 : :   218 : 1545 : SDDB \*pSDDB = SDDB::instance();  219 : 1545 : try  220 : : {  221 : 1545 : pSDDB->EPReadComplete(nrctx->NumEP);  222 : : }  223 [ # # ]: 0 : catch( DPL::Error \*pChild )  224 : : {  225 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  226 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  227 : : ErrorHandler::eErrFunction,  228 : : pErr->getErrorMsg(ErrorHandler::eErrFunction,  229 : : "SDDB::EPReadComplete"));  230 : : }  231 : :   232 : :   233 : 1545 : pgc->current += sizeof(vsbPH1) + SizeOfOption[ph1.isOptionAdded] ;  234 : : }  235 : :   236 : :   237 [ + + ]: 5953 : if(nrctx->CurrEPNum == nrctx->NumEP)  238 : : {  239 : 457 : nrctx->isEmpty = true ;  240 : 457 : return -1 ;  241 : : }   242 : :   243 : 5496 : SkipSize = vsbBufferToEP(pgc->current, nrctx->NumFieldsInEP, pgc->X0Y0WordLength, nrctx->L1WordLength, nrctx->L2WordLength, ep) ;  244 : 5496 : ep->PatCode = nrctx->PatCode ;  245 : :   246 : 5496 : \*Nx = \*Ny = 1 ;  247 : 5496 : \*Xe = ep->X0 ;  248 : 5496 : \*Ye = ep->Y0 ;  249 : :   250 : 5496 : pgc->ppc.ep = \*ep ;  251 : :   252 : : // Copy pre-translated EP to CRContext  253 [ + + ]: 5496 : if ( Composite == pgc->ppc.WhichContext)  254 : : {  255 : 79 : pgc->ppc.crctx.ReprEP = \*ep ;  256 : 79 : pgc->ppc.crctx.ReprNx = \*Nx ;  257 : 79 : pgc->ppc.crctx.ReprNy = \*Ny ;  258 : 79 : pgc->ppc.crctx.ReprXe = \*Xe ;  259 : 79 : pgc->ppc.crctx.ReprYe = \*Ye ;  260 : : }  261 [ + + ][ - + ]: 5417 : else if ( 1.0F != pgc->TranslateFactor ||  262 : : pgc->mirror)   263 : : {  264 [ - + ]: 40 : if( 0 > TranslateEP(pgc, ep, Xe, Ye) )  265 : 0 : return -1 ;  266 : : }  267 : :   268 : 5496 : nrctx->CurrEPNum++ ;  269 : 5496 : pgc->current += SkipSize ;  270 : :   271 : 5953 : return 0 ;  272 : : }  273 : :   274 : 1215 : } //namespace nft |

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