| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [LibVsb/include](http://docs.google.com/index.html) - VSBCellRRIDTracker.h (source / [functions](http://docs.google.com/VSBCellRRIDTracker.h.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 15 | 17 | 88.2 % | | Date: | 2012-03-23 |  | Functions: | 0 | 0 | - | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 12 | 12 | 100.0 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   2 : : \* Copyrighted 2007 by Interra Systems, USA  3 : : \* All rights reserved   4 : : \*  5 : : \* Date of Creation: Fri Jul 20 13:12:07 IST 2007  6 : : \* File Name: VSBCellRRIDTracker.h  7 : : \* Purpose:   8 : : \* Author:   9 : : \*  10 : : \* Revision History:   11 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/   12 : : #ifndef \_\_VSBCELLRRIDTRACKER\_H\_\_   13 : : #define \_\_VSBCELLRRIDTRACKER\_H\_\_  14 : :   15 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  16 : : \* Header Include section   17 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  18 : : #include <vector>  19 : : using namespace std;  20 : :   21 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  22 : : \* Class Declarations section   23 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/   24 : : namespace nft{  25 : :   26 : : class VSBCellRRIDTracker  27 : : {  28 : : private:  29 : : vector<unsigned char> mCountEncodeVc;  30 : : private:  31 : : // Copy Constructor Blocked  32 : : VSBCellRRIDTracker(const VSBCellRRIDTracker&);  33 : :   34 : : // Assignment operator Blocked  35 : : VSBCellRRIDTracker& operator = (const VSBCellRRIDTracker&);  36 : : public:  37 : : VSBCellRRIDTracker()  38 : 2188 : {}  39 : : void reInit()  40 : 1094 : {  41 : 1094 : mCountEncodeVc.clear();  42 : 1094 : mCountEncodeVc.resize(100); //Enumerate this magic number  43 : : //Needed to avoid SEGSIGV when we   44 : : //access mCountEncodeVc[0] first time  45 : : }   46 : : //void dumpContents();  47 : :   48 : : unsigned short getCountEncodeVc(uint32\_t idx)  49 [ + + ][ + + ]: 1261 : {  50 : 1261 : if (idx > mCountEncodeVc.size())  51 : 0 : {  52 : : return 0;  53 : : }  54 : : else  55 : 2522 : {  56 : : return mCountEncodeVc[idx];  57 : : }  58 : :   59 : : }  60 : : unsigned short operator[] (uint32\_t idx) //readonly  61 : : {  62 : : if (idx > mCountEncodeVc.size())  63 : : {  64 : : return 0;  65 : : }  66 : : else  67 : : {  68 : : return mCountEncodeVc[idx];  69 : : }  70 : : }  71 : :   72 : : unsigned short incrCountEncodeVc(uint32\_t idx)  73 [ + + ][ + + ]: 2374 : {  74 : : #ifdef DEVEL\_DEBUG  75 : : cout <<"RRID = "<<idx<<endl;  76 : : cout<<"size = "<<mCountEncodeVc.size()<<endl;  77 : : cout<<"size = "<<mCountEncodeVc.capacity()<<endl;  78 : : #endif  79 : : //Ensures that count lies in [0, 4]  80 : : //Logic of increment: 0->1->4  81 : : /\*   82 : : cout<<"SIZE = "<<mCountEncodeVc.size();  83 : :   84 : : cout<<"CAPACITY = "<<mCountEncodeVc.capacity();  85 : : \*/  86 : :   87 : 1261 : if (idx >= mCountEncodeVc.size())  88 : : {  89 : 0 : mCountEncodeVc.resize(idx<<1);   90 : : }  91 : : else  92 : : {  93 : : //We have enough space  94 : : }   95 : : //We rely on the initialization of the vector.  96 : : //All chars in the vector are initialized to zero  97 : 1261 : unsigned short oldVal = mCountEncodeVc[idx];  98 [ + + ]: 1261 : if ( oldVal >= 4)  99 : : {  100 : : //cout<<"Keeping to limit"<<endl;  101 : : }  102 [ + + ]: 1192 : else if (oldVal == 1)  103 : : {  104 : 76 : mCountEncodeVc[idx] = 4; //reference cell  105 : : }   106 : : else //(oldVal == 0)  107 : : {  108 : 1116 : mCountEncodeVc[idx] = 1;  109 : 2374 : }  110 : : return oldVal;  111 : :   112 : : }  113 : : unsigned short decrCountEncodeVc(uint32\_t idx)  114 : : {  115 : : unsigned short oldVal = mCountEncodeVc[idx];  116 : : if (oldVal == 4) //reference cell  117 : : {  118 : : mCountEncodeVc[idx] = 2; //semi processed reference cell  119 : : }  120 : : else if (oldVal == 1) //value cell  121 : : {  122 : : mCountEncodeVc[idx] = 0; //semi processed value cell  123 : : }  124 : : else //(oldVal == 2 || oldVal == 0 )  125 : : {  126 : : //nothing to change  127 : : }  128 : : return oldVal;  129 : : }   130 : : };   131 : :   132 : : } //namespace nft  133 : : #endif |

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