| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [Common/src](http://docs.google.com/index.html) - LayerCalculator.cpp (source / [functions](http://docs.google.com/LayerCalculator.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 170 | 186 | 91.4 % | | Date: | 2012-03-23 |  | Functions: | 8 | 9 | 88.9 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 14 | 30 | 46.7 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : :   2 : : #include <math.h>  3 : : #include "LayerCalculator.h"  4 : : #include "RetryFunc.h"  5 : : #include "vchip.h"  6 : :   7 : : namespace nft{  8 : :   9 : : namespace \_impl {  10 : :   11 : : struct LayerInfo {  12 : : uint32\_t mSFLNo;  13 : : uint32\_t mSFLID;  14 : : int64\_t mSFLOriginX; // [AU]  15 : : int64\_t mSFLOriginY; // [AU]  16 : : uint32\_t mPreCalcSFCount;  17 : : int64\_t mTFOriginX; // [AU]  18 : : int64\_t mTFOriginY; // [AU]  19 : : uint32\_t mPreCalcTFCount;  20 : : };  21 : : }  22 : :   23 : : LayerCalculator::LayerCalculator(  24 : : DPMANAGER\_MESSAGE\_CONVERTER \*msgConv,  25 : : const LAYERPARAMLIST \*layerParamList)  26 : 1119 : :mMsgConv(msgConv),  27 : : mLayerParamList(layerParamList),  28 : : mLayerSets(),  29 : 2238 : mNextLayerSet() {  30 : : }  31 : :   32 : 2238 : LayerCalculator::~LayerCalculator() {  33 : 1119 : for( std::vector<LayerSet \*>::iterator lset = mLayerSets.begin();  34 : : lset != mLayerSets.end(); lset++ )  35 : : {  36 : 1128 : LayerSet \*layers = \*lset;  37 : 1128 : for( LAYERPARAMLIST::iterator l = layers->mLayers.begin();  38 : : l != layers->mLayers.end(); ++l)  39 : : {  40 : 1197 : delete (\*l);  41 : : }  42 [ + - ][ # # ]: 1128 : delete layers;  43 : : }  44 : : }  45 : :   46 : : bool  47 : 1119 : LayerCalculator::calculateLayers( ChipCnfFileParams \*inChipCnf ) {  48 [ + + ]: 1119 : if( mMsgConv->usFrameOrientation == 1 ) {  49 : 1118 : return calculateLayersForHorizontalDPB( inChipCnf );  50 : : }  51 : 1119 : return calculateLayersForVerticalDPB( inChipCnf );  52 : : }  53 : :   54 : : bool  55 : : LayerCalculator::calculateLayersForHorizontalDPB(  56 : 1118 : ChipCnfFileParams \*inChipCnf ) {  57 : 1118 : uint32\_t numFrames = inChipCnf->get\_uiChipFrms\_();  58 : 1118 : std::vector<int64\_t> originalOriginSFX;  59 : 1118 : std::vector<int64\_t> originalOriginTFX;  60 : 1118 : std::vector<\_impl::LayerInfo \*> currentStatus;  61 : 1118 : for( LAYERPARAMLIST::const\_iterator layer = mLayerParamList->begin();  62 : : layer != mLayerParamList->end(); layer++ )  63 : : {  64 [ + - ]: 1187 : \_impl::LayerInfo \*info = new(std::nothrow) \_impl::LayerInfo();  65 [ - + ]: 1187 : if( !info )  66 : : {  67 : 0 : return false;  68 : : }  69 : 1187 : DPMANAGER\_MESSAGE\_SFL\_INFO \*pSFL = \*layer;  70 : 1187 : info->mSFLNo = pSFL->uiSFLNo;  71 : 1187 : info->mSFLID = pSFL->uiSFLID;  72 : 1187 : info->mSFLOriginX = pSFL->llSFLOriginX;  73 : 1187 : info->mSFLOriginY = pSFL->llSFLOriginY;  74 : 1187 : info->mTFOriginX = pSFL->llTFOriginX;  75 : 1187 : info->mTFOriginY = pSFL->llTFOriginY;  76 : 1187 : currentStatus.push\_back( info );  77 : 1187 : originalOriginSFX.push\_back( pSFL->llSFLOriginX );  78 : 1187 : originalOriginTFX.push\_back( pSFL->llTFOriginX );  79 : : }  80 : 1118 : int32\_t sfSizeX = mMsgConv->iSFSizeX;  81 : 1118 : int32\_t sfSizeY = mMsgConv->iSFSizeY;  82 : 1118 : int32\_t tfSizeX = mMsgConv->iTFSizeX;  83 : 1118 : int32\_t tfSizeY = mMsgConv->iTFSizeY;  84 : 1118 : uint32\_t frameSpanY = 0;  85 [ + - ][ + + ]: 2245 : for( size\_t j = 0; j < numFrames; j++ )  86 : : {  87 : 1127 : uint32\_t numSubFrames = inChipCnf->get\_frameSubframeCount( j );  88 : : uint32\_t frameHeight = inChipCnf->get\_frameHeight( 0, j )  89 : 1127 : \* inChipCnf->get\_llChipBlkSizeY\_();  90 : 1127 : frameSpanY += frameHeight;  91 : 2323 : for( size\_t layerCount = 0; layerCount < mLayerParamList->size();  92 : : layerCount ++ )  93 : : {  94 : 1196 : \_impl::LayerInfo \*info = currentStatus[ layerCount ];  95 : : {  96 : 1196 : int64\_t spanY = frameSpanY - info->mSFLOriginY;  97 : 1196 : uint32\_t sfCountY = spanY / sfSizeY;  98 : 1196 : sfCountY += ( sfCountY \* sfSizeY < spanY ) ? 1 : 0 ;  99 : 1196 : info->mPreCalcSFCount = sfCountY;  100 : : }  101 : : {  102 : 1196 : int64\_t spanY = frameSpanY - info->mTFOriginY;  103 : 1196 : uint32\_t tfCountY = spanY / tfSizeY;  104 : 1196 : tfCountY += ( tfCountY \* tfSizeY < spanY ) ? 1 : 0 ;  105 : 1196 : info->mPreCalcTFCount = tfCountY;  106 : : }  107 : : }  108 : 1127 : uint32\_t i = 0;  109 : 1127 : uint32\_t frameSpanX = 0;  110 : 1127 : do  111 : : {  112 : : uint32\_t frameWidth = inChipCnf->get\_frameWidth( i, j )  113 : 1127 : \* inChipCnf->get\_llChipBlkSizeX\_();  114 : 1127 : LayerSet \*layers = new LayerSet();  115 : 1127 : mLayerSets.push\_back( layers );  116 : 1127 : frameSpanX += frameWidth;  117 : 2323 : for( size\_t layerCount = 0; layerCount < mLayerParamList->size();  118 : : layerCount ++ )  119 : : {  120 : 1196 : \_impl::LayerInfo \*info = currentStatus[ layerCount ];  121 : : DPMANAGER\_MESSAGE\_SFL\_INFO \*pSFL  122 : 1196 : = new DPMANAGER\_MESSAGE\_SFL\_INFO();  123 : 1196 : layers->mLayers.push\_back( pSFL );  124 : 1196 : pSFL->uiSFLID = info->mSFLID;  125 : 1196 : pSFL->uiSFLNo = info->mSFLNo;  126 : 1196 : pSFL->llSFLOriginX = info->mSFLOriginX;  127 : 1196 : pSFL->llSFLOriginY = info->mSFLOriginY;  128 : 1196 : pSFL->llTFOriginX = info->mTFOriginX;  129 : 1196 : pSFL->llTFOriginY = info->mTFOriginY;  130 : : {  131 : 1196 : int64\_t spanX = frameSpanX - info->mSFLOriginX;  132 : 1196 : uint32\_t sfCountX = spanX / sfSizeX;  133 : 1196 : sfCountX += ( sfCountX \* sfSizeX < spanX ) ? 1 : 0 ;  134 : 1196 : pSFL->uiSFCountX = sfCountX;  135 : 1196 : pSFL->uiSFCountY = info->mPreCalcSFCount;  136 : 1196 : info->mSFLOriginX += sfCountX \* sfSizeX;  137 : : }  138 : :   139 : : {  140 : 1196 : int64\_t spanX = frameSpanX - info->mTFOriginX;  141 : 1196 : uint16\_t tfCountX = spanX / tfSizeX;  142 : 1196 : tfCountX += ( tfCountX \* tfSizeX < spanX ) ? 1 : 0 ;  143 : 1196 : pSFL->usTFCountX = tfCountX;  144 : 1196 : pSFL->usTFCountY = info->mPreCalcTFCount;  145 : 1196 : info->mTFOriginX += tfCountX \* tfSizeX;  146 : : }  147 : : }  148 : 1127 : i++;  149 [ - + ]: 1127 : }  150 : : while( i < numSubFrames );  151 : :   152 : 2323 : for( size\_t layerCount = 0; layerCount < mLayerParamList->size();  153 : : layerCount ++ )  154 : : {  155 : 1196 : \_impl::LayerInfo \*info = currentStatus[ layerCount ];  156 : 1196 : info->mSFLOriginX = originalOriginSFX[ layerCount ];  157 : 1196 : info->mSFLOriginY += info->mPreCalcSFCount \* sfSizeY;  158 : 1196 : info->mTFOriginX = originalOriginTFX[ layerCount ];  159 : 1196 : info->mTFOriginY += info->mPreCalcTFCount \* tfSizeY;  160 : : }  161 : : }  162 : 1118 : for( std::vector<\_impl::LayerInfo \*>::iterator it = currentStatus.begin();  163 : : it != currentStatus.end(); ++it )  164 : : {  165 : 1187 : delete \*it;  166 : : }  167 : 1118 : mNextLayerSet = mLayerSets.begin();  168 : 1118 : return true;  169 : : }  170 : :   171 : : bool  172 : 1 : LayerCalculator::calculateLayersForVerticalDPB( ChipCnfFileParams \*inChipCnf ) {  173 : 1 : uint32\_t numFrames = inChipCnf->get\_uiChipFrms\_();  174 : 1 : std::vector<int64\_t> originalOriginSFY;  175 : 1 : std::vector<int64\_t> originalOriginTFY;  176 : 1 : std::vector<\_impl::LayerInfo \*> currentStatus;  177 : 1 : for( LAYERPARAMLIST::const\_iterator layer = mLayerParamList->begin();  178 : : layer != mLayerParamList->end(); layer++ )  179 : : {  180 [ + - ]: 1 : \_impl::LayerInfo \*info = new(std::nothrow) \_impl::LayerInfo();  181 [ - + ]: 1 : if( !info )  182 : : {  183 : 0 : return false;  184 : : }  185 : 1 : DPMANAGER\_MESSAGE\_SFL\_INFO \*pSFL = \*layer;  186 : 1 : info->mSFLNo = pSFL->uiSFLNo;  187 : 1 : info->mSFLID = pSFL->uiSFLID;  188 : 1 : info->mSFLOriginX = pSFL->llSFLOriginX;  189 : 1 : info->mSFLOriginY = pSFL->llSFLOriginY;  190 : 1 : info->mTFOriginX = pSFL->llTFOriginX;  191 : 1 : info->mTFOriginY = pSFL->llTFOriginY;  192 : 1 : currentStatus.push\_back( info );  193 : 1 : originalOriginSFY.push\_back( pSFL->llSFLOriginY );  194 : 1 : originalOriginTFY.push\_back( pSFL->llTFOriginY );  195 : : }  196 : 1 : int32\_t sfSizeX = mMsgConv->iSFSizeX;  197 : 1 : int32\_t sfSizeY = mMsgConv->iSFSizeY;  198 : 1 : int32\_t tfSizeX = mMsgConv->iTFSizeX;  199 : 1 : int32\_t tfSizeY = mMsgConv->iTFSizeY;  200 : 1 : uint32\_t frameSpanX = 0;  201 [ + - ][ - + ]: 2 : for( size\_t j = 0; j < numFrames; j++ )  202 : : {  203 : 1 : uint32\_t numSubFrames = inChipCnf->get\_frameSubframeCount( j );  204 : : uint32\_t frameWidth = inChipCnf->get\_frameHeight( 0, j )  205 : 1 : \* inChipCnf->get\_llChipBlkSizeX\_();  206 : 1 : frameSpanX += frameWidth;  207 : 2 : for( size\_t layerCount = 0; layerCount < mLayerParamList->size();  208 : : layerCount ++ )  209 : : {  210 : 1 : \_impl::LayerInfo \*info = currentStatus[ layerCount ];  211 : : {  212 : 1 : int64\_t spanX = frameSpanX - info->mSFLOriginX;  213 : 1 : uint32\_t sfCountX = spanX / sfSizeX;  214 : 1 : sfCountX += ( sfCountX \* sfSizeX < spanX ) ? 1 : 0 ;  215 : 1 : info->mPreCalcSFCount = sfCountX;  216 : : }  217 : : {  218 : 1 : int64\_t spanX = frameSpanX - info->mTFOriginX;  219 : 1 : uint32\_t tfCountX = spanX / tfSizeX;  220 : 1 : tfCountX += ( tfCountX \* tfSizeX < spanX ) ? 1 : 0 ;  221 : 1 : info->mPreCalcTFCount = tfCountX;  222 : : }  223 : : }  224 : 1 : uint32\_t i = 0;  225 : 1 : uint32\_t frameSpanY = 0;  226 : 1 : do  227 : : {  228 : : uint32\_t frameHeight = inChipCnf->get\_frameWidth( i, j )  229 : 1 : \* inChipCnf->get\_llChipBlkSizeX\_();  230 : 1 : LayerSet \*layers = new LayerSet();  231 : 1 : mLayerSets.push\_back( layers );  232 : 1 : frameSpanY += frameHeight;  233 : 2 : for( size\_t layerCount = 0; layerCount < mLayerParamList->size();  234 : : layerCount ++ )  235 : : {  236 : 1 : \_impl::LayerInfo \*info = currentStatus[ layerCount ];  237 : : DPMANAGER\_MESSAGE\_SFL\_INFO \*pSFL  238 : 1 : = new DPMANAGER\_MESSAGE\_SFL\_INFO();  239 : 1 : layers->mLayers.push\_back( pSFL );  240 : 1 : pSFL->uiSFLID = info->mSFLID;  241 : 1 : pSFL->uiSFLNo = info->mSFLNo;  242 : 1 : pSFL->llSFLOriginX = info->mSFLOriginX;  243 : 1 : pSFL->llSFLOriginY = info->mSFLOriginY;  244 : 1 : pSFL->llTFOriginX = info->mTFOriginX;  245 : 1 : pSFL->llTFOriginY = info->mTFOriginY;  246 : : {  247 : 1 : int64\_t spanY = frameSpanY - info->mSFLOriginY;  248 : 1 : uint32\_t sfCountY = spanY / sfSizeY;  249 : 1 : sfCountY += ( sfCountY \* sfSizeY < spanY ) ? 1 : 0 ;  250 : 1 : pSFL->uiSFCountX = info->mPreCalcSFCount;  251 : 1 : pSFL->uiSFCountY = sfCountY;  252 : 1 : info->mSFLOriginY += sfCountY \* sfSizeY;  253 : : }  254 : :   255 : : {  256 : 1 : int64\_t spanY = frameSpanY - info->mTFOriginY;  257 : 1 : uint16\_t tfCountY = spanY / tfSizeY;  258 : 1 : tfCountY += ( tfCountY \* tfSizeY < spanY ) ? 1 : 0 ;  259 : 1 : pSFL->usTFCountX = info->mPreCalcTFCount;  260 : 1 : pSFL->usTFCountY = tfCountY;  261 : 1 : info->mTFOriginY += tfCountY \* tfSizeY;  262 : : }  263 : : }  264 : 1 : i++;  265 [ - + ]: 1 : }  266 : : while( i < numSubFrames );  267 : :   268 : 2 : for( size\_t layerCount = 0; layerCount < mLayerParamList->size();  269 : : layerCount ++ )  270 : : {  271 : 1 : \_impl::LayerInfo \*info = currentStatus[ layerCount ];  272 : 1 : info->mSFLOriginX += info->mPreCalcSFCount \* sfSizeX;  273 : 1 : info->mSFLOriginY = originalOriginSFY[ layerCount ];  274 : 1 : info->mTFOriginX += info->mPreCalcTFCount \* tfSizeX;  275 : 1 : info->mTFOriginY = originalOriginTFY[ layerCount ];  276 : : }  277 : : }  278 : 1 : for( std::vector<\_impl::LayerInfo \*>::iterator it = currentStatus.begin();  279 : : it != currentStatus.end(); ++it )  280 : : {  281 : 1 : delete \*it;  282 : : }  283 : 1 : mNextLayerSet = mLayerSets.begin();  284 : 1 : return true;  285 : : }  286 : :   287 : : bool  288 : : LayerCalculator::getNextLayerSet( int64\_t inDpbX, int64\_t inDpbY,  289 : 0 : const LAYERPARAMLIST \*layerParamList ) {  290 : 0 : if( mNextLayerSet == mLayerSets.end() )  291 : : {  292 : 0 : return false; //Should not happen  293 : : }  294 : 0 : LayerSet \*pLayers = \*mNextLayerSet;  295 [ # # ][ # # ]: 0 : for( size\_t layerCount = 0; layerCount < mMsgConv->uiSFLCountInDpb;  296 : : layerCount++ )  297 : : {  298 : : DPMANAGER\_MESSAGE\_SFL\_INFO \*pTarget = \*(layerParamList->begin()  299 : 0 : + layerCount);  300 : : DPMANAGER\_MESSAGE\_SFL\_INFO \*pSrc = \*(pLayers->mLayers.begin()  301 : 0 : + layerCount);  302 : 0 : ::memcpy( pTarget, pSrc, sizeof( DPMANAGER\_MESSAGE\_SFL\_INFO ) );  303 : 0 : pTarget->llSFLOriginX -= inDpbX;  304 : 0 : pTarget->llSFLOriginY -= inDpbY;  305 : 0 : pTarget->llTFOriginX -= inDpbX;  306 : 0 : pTarget->llTFOriginY -= inDpbY;  307 : : }  308 : 0 : mNextLayerSet++;  309 : 0 : return true;  310 : : }  311 : :   312 : :   313 : 1215 : }; |

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