```
/*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*
 *
 * File Name   : MSviPreprocessBlock_06Std260.v
 * Description : Pre-procee Block for SVI-06
 * Simulator   : VeritakWin Ver.3.84D Build May.23.2011
 * Implem. Tool: ISE Webpack Ver.13.2 Appl Ver. 0.61xd
 * Explanation : Rev.1.0
 *
 * Copyright(c) 2008-2015, by Net-Vision Corp. All rights reserved.
 * (Note) For this source code, it is forbidden using and issuing
 *        without permission.
 * （注） このソース・コードの無断使用および無断持ち出しを禁止します.
 *
 * Revision    :
 * 21/Feb-2014 06Std260-Renewal Release...                               K.Kudo
 *
 *~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*~`^*/
//-----------------------------------------------------------------------------
module MSviPreprocessBlock
#(
parameter       pDebug  = "no"  // ChipScopeによるデバッグ用途
)(
//-----------------------------------------------------------------------------
// General CAM-PORT Source Side from External Port
// Camera Standard-I/F
input [15:0]    iCamDtX,
input           iCamHsX,
input           iCamVsX,
input           iCamClkX,
//
// Camera I2C-I/F
input           iCamSclRsX,     // SCL Reference Line from I2C-Slave side
output          oCamSclZsX,     // SCL High-Z Control Line for I2C-Slave side
input           iCamSdaRsX,     // SDA Reference Line from I2C-Slave side
output          oCamSdaZsX,     // SDA High-Z Control Line for I2C-Slave side
//
// Camera System-Out
input           iCamXRSTRefX,
output          oCamXRSTCtlX,
output          oClkOutX,
//
// General Purpose-I/O
output[ 7:0]    oGpOutX,
input [ 7:0]    iGpInX,
//
//-----------------------------------------------------------------------------
// General CAM-PORT Destination Side for Internal Blockt
// Camera Standard-I/F
output[15:0]    oCamDtI,
output          oCamHsI,
output          oCamVsI,
output          oCamClkI,
//
// Camera I2C-I/F
output          oCamSclRsI,     // SCL Reference Line from I2C-Slave side
input           iCamSclZsI,     // SCL High-Z Control Line for I2C-Slave side
output          oCamSdaRsI,     // SDA Reference Line from I2C-Slave side
input           iCamSdaZsI,     // SDA High-Z Control Line for I2C-Slave side
//
// Camera System-Out
output          oCamXRSTRefI,
input           iCamXRSTCtlI,
input           iClkOutPllI,    // External Pll-Clock
input [ 1:0]    inClkOutDivI,   // External Pulluped Dip-SW Async. Input
input           inClkOutEnaI,   // Async Neg. Clock Output Enable
//
// General Purpose-I/O
input [ 7:0]    iGpOutI,
output[ 7:0]    oGpInI,
//
//-----------------------------------------------------------------------------
// SVI-Internal Mode and Control Signals Reference
input           iCamClkInvert,
input           iSyncPolarity,
input           iCap16EnaS,
```

```
 76:output          oCap16EnaD,
 77:input [1:0]     iMonCapDoneS,   // Mon_Cap-Block Done Src-Side, Async. Each Clock
 78:output[1:0]     oMonCapDoneD,   // Mon_Cap-Block Done Dst-Side, Async. Each Clock
 79://
 80://---------------------------------------------------------------------------
 81:// Block System Ports
 82:// General Control/Status
 83:output[63:0]    oBSTTS,
 84:input [63:0]    iBCTRL,
 85:input [ 7:0]    iBCWRP,         // BCTRL Write Pulse for Each Byte Lane
 86://
 87:// Common System Lines
 88:input           iBRST,          // Block System Reset(Positive Sync.)
 89:input           iBCLK           // Block System Clock
 90://
 91:);
 92:
 93://---------------------------------------------------------------------------
 94:// System Line Declaration
 95://---------------------------------------------------------------------------
 96:  wire          wCCLKS;     // Camera Clock Source Side
 97:  wire          wCRSTS;     // wCCLKS Side Sync.  Pos. Reset
 98:  //
 99:  BUFG  mBufgCamClk(    .O(wCCLKS),     .I(iCamClkX)    );  // Target Sensor Clock
100:  assign        oCamClkI    = wCCLKS;
101:
102:  // Clock-Out Port
103:  MCamClkOutDivider
104:    mCamClkOutDivider(
105:        .oClkOutX(oClkOutX),            .iClkOutI(iClkOutPllI),
106:        .inClkOutDivI(inClkOutDivI),    .inClkOutEnaI(inClkOutEnaI),
107:        .inRST(~iBRST)  // inCRSTであってはならない。
108:    );                  // inCRST解除前にカメラへのクロック供給は安定していなければならない。
109:
110://---------------------------------------------------------------------------
111:// IDDR2 Part, Input 1st. Regs
112://---------------------------------------------------------------------------
113:  wire[15:0]    wCamDtS;
114:  wire          wCamHsS,    wCamVsS;
115:  reg           rCkInv;
116:  genvar    gvn;
117:  generate
118:    for ( gvn = 0;  gvn < 16; gvn = gvn + 1 )
119:    begin :lEachClkEdgeAlignmentPortDt
120:      MClkEdgeAlignmentPort
121:        mClkEdgeAlignmentPortDt(
122:            .oQI(wCamDtS[gvn]), .iDX(iCamDtX[gvn]),
123:            .iCKINV(rCkInv),    .iCLKG(wCCLKS)
124:        );
125:    end
126:  endgenerate
127:
128:  MClkEdgeAlignmentPort
129:    mClkEdgeAlignmentPortHs(
130:        .oQI(wCamHsS),      .iDX(iCamHsX),
131:        .iCKINV(rCkInv),    .iCLKG(wCCLKS)
132:    );
133:
134:  MClkEdgeAlignmentPort
135:    mClkEdgeAlignmentPortVs(
136:        .oQI(wCamVsS),      .iDX(iCamVsX),
137:        .iCKINV(rCkInv),    .iCLKG(wCCLKS)
138:    );
139:
140:  always @( posedge wCCLKS )
141:  begin
142:    rCkInv  <= iCamClkInvert;
143:  end
144:
145://---------------------------------------------------------------------------
146:  // 外部同期信号入力を指定の極性をもとに共通化する。
147:  // 前処理ブロック内部では、Sync-Mode="Active Low"を想定。
148:  wire          wCamHsSspc  = iSyncPolarity ^ wCamHsS;
149:  wire          wCamVsSspc  = iSyncPolarity ^ wCamVsS;
150:  wire          wHsInvert,  wVsInvert;
```



```
151:  wire          wCamHsSsrc  = wHsInvert     ^ wCamHsSspc;
152:  wire          wCamVsSsrc  = wVsInvert     ^ wCamVsSspc;
153:  // 従来のiSyncPolarityによるSyncModeの設定に加え、VS/HS個別反転機能を追加。
154:  // 下位互換のために単純に上記の実装とする。
155:
156://---------------------------------------------------------------------------
157:// 第一段階として、水平ライン毎に間引くことによって、垂直方向の削減をする。
158:// 単純にHSyncをゲートすることによって間引く
159://---------------------------------------------------------------------------
160:  wire[15:0]    wCamDtVH;
161:  wire          wCamVsVH;
162:  wire          wCamHsVH;
163:  wire          wENAV;
164:  wire[ 3:0]    wRDCV;
165:  MCamVertReducer
166:    mCamVertReducer(
167:        // General CAM-PORT Source Side
168:        .iCamDtS(wCamDtS),  .iCamVsS(wCamVsSsrc),   .iCamHsS(wCamHsSsrc),
169:        // General CAM-PORT Destination Side
170:        .oCamDtD(wCamDtVH), .oCamVsD(wCamVsVH),     .oCamHsD(wCamHsVH),
171:        // Control/Status Port
172:        .iENA(wENAV),       .iRDCV(wRDCV),
173:        // Common System Lines
174:        .iRST(wCRSTS),      .iCLK(wCCLKS)
175:    );
176:
177://---------------------------------------------------------------------------
178:// 第二段階として、水平ライン中のピクセルを間引く。
179:// 最終的な出力として、有効なHSyncアサート中、連続な有効画素データを出力する必要があるので、
180:// ラインバッファを持って、一ラインつめてから、出力するようにする。
181://---------------------------------------------------------------------------
182:  wire          wENAH;
183:  wire[ 3:0]    wRDCH;
184:  wire[ 1:0]    wCPPXL;
185:  wire          wChrCamVsD, wChrCamHsD;
186:  MCamHorzReducer
187:    mCamHorzReducer(
188:        // General CAM-PORT Source Side
189:        .iCamDtS(wCamDtVH), .iCamVsS(wCamVsVH), .iCamHsS(wCamHsVH),
190:        // General CAM-PORT Destination Side
191:        .oCamDtD(oCamDtI),  .oCamVsD(wChrCamVsD),   .oCamHsD(wChrCamHsD),
192:        // Control/Status Port
193:        .iENA(wENAH),       .iRDCH(wRDCH),  .iCPPXL(wCPPXL),
194:        // Common System Lines
195:        .iRST(wCRSTS),      .iCLK(wCCLKS)
196:    );
197:
198:  // '13-07/08: iSyncPolarityで反転入力した場合は、
199:  // 戻して出力する必要がある。
200:  assign    oCamHsI = iSyncPolarity ^ wChrCamHsD;
201:  assign    oCamVsI = iSyncPolarity ^ wChrCamVsD;
202:
203://---------------------------------------------------------------------------
204:// 前処理ブロック内部のリセット・システムを構成する。
205://---------------------------------------------------------------------------
206:  wire          wBRstFw;    // Block Internal Reset by FW Control
207:  wire          wnBRstI = ~(iBRST | wBRstFw);
208:  (* KEEP_HIERARCHY=pDebug *)
209:  MResetUnit
210:    #(  .pRstSUse("yes"),       .pRstSCnt(15),
211:        .pRstAUse("no"),        .pRstACnt(20),
212:        .pSequence("Each"),     .pTPlsUse("no") )
213:    mCRst(
214:        .oRSTS(wCRSTS),         .onRSTA(),          .iCLK(wCCLKS),
215:        .inRSTX(iCamXRSTRefX),  .inRSTI(wnBRstI),   .iTPLS(1'b1)
216:    );
217:
218://---------------------------------------------------------------------------
219:// 前処理ブロックのI/O、Control/Status空間を構成する。
220://---------------------------------------------------------------------------
221:  assign        wBRstFw         = iBCTRL[0];        // Block Internal Reset by SW Control
222:  assign        wENAV           = iBCTRL[7];
223:  assign        wRDCV           = iBCTRL[15:12];
224:  assign        wENAH           = iBCTRL[3];
225:  assign        wRDCH           = iBCTRL[11: 8];
```



```
226:  assign        wCPPXL          = iBCTRL[2:1];
227:  assign        wHsInvert       = iBCTRL[5];
228:  assign        wVsInvert       = iBCTRL[6];
229:  // iBCTRLはiBCLKに同期化されている。
230:  // この前処理ブロックでのパラメータ渡しでは、
231:  // 動作中は固定的なので同期化の必要なしとしてそのままの接続で実装する。
232:
233:  reg [63: 0]   rBstts;         assign  oBSTTS[63: 0]   = rBstts;
234:  always @( posedge iBCLK )
235:  begin
236:    rBstts[63:0]    <= iBCTRL[63:0];
237:  end
238:
239://---------------------------------------------------------------------------
240:// 未使用の出力ポート処理
241://---------------------------------------------------------------------------
242:  assign    oCamSclRsI      = iCamSclRsX;
243:  assign    oCamSclZsX      = iCamSclZsI;
244:  assign    oCamSdaRsI      = iCamSdaRsX;
245:  assign    oCamSdaZsX      = iCamSdaZsI;
246:  assign    oCamXRSTRefI    = iCamXRSTRefX;
247:  assign    oCamXRSTCtlX    = iCamXRSTCtlI;
248:  assign    oGpOutX         = iGpOutI;
249:  assign    oGpInI          = iGpInX;
250:  assign    oCap16EnaD      = iCap16EnaS;
251:  assign    oMonCapDoneD    = iMonCapDoneS;
252:
253:  // <51>:P6 <-oGpOutX[3]    |   <52>:P7 <-oGpOutX[4]
254:  // <53>:P8 <-oGpOutX[5]    |   <54>:P9 <-oGpOutX[6]
255:  // <55>:P10<-oGpOutX[7]    |   <56>:P11->iGpInX [3]
256:  // <57>:P12->iGpInX [4]    |   <58>:P13->iGpInX [5]
257:  // <59>:P14->iGpInX [6]    |   <60>:P15->iGpInX [7]
258:  // <01>:VDD_L              |   <02>:GND
259:  // <03>:P0 ->iGpInX [0]    |   <04>:GND
260:  // <05>:P1 ->iGpInX [1]    |   <06>:GND
261:  // <07>:P2 ->iGpInX [2]    |   <08>:GND
262:  // <09>:P3 <-oGpOutX[0]    |   <10>:GND
263:  // <11>:P4 <-oGpOutX[1]    |   <12>:HS
264:  // <13>:VS                 |   <14>:XRST
265:  // ...                         ...
266:  // <49>:3.3V               |   <50>:P5 <-oGpOutX[2]
267:
268://---------------------------------------------------------------------------
269:endmodule   // MSviPreprocessBlock
270://---------------------------------------------------------------------------
271://---------------------------------------------------------------------------
272:// 垂直水平方向に関する間引きモジュール
273://---------------------------------------------------------------------------
274:module MCamVertReducer
275:(
276://---------------------------------------------------------------------------
277:// General CAM-PORT Source Side
278://
279:input [15:0]    iCamDtS,
280:input           iCamVsS,
281:input           iCamHsS,
282://
283://---------------------------------------------------------------------------
284:// General CAM-PORT Destination Side
285://
286:output[15:0]    oCamDtD,
287:output          oCamVsD,
288:output          oCamHsD,
289://
290://---------------------------------------------------------------------------
291:// Control/Status Port
292://
293:input           iENA,       // 前処理機能のイネーブル
294:input [ 3:0]    iRDCV,      // Reduce-V
295://
296://---------------------------------------------------------------------------
297:// Common System Lines
298:// カメラのデータ・クロックと同じクロック。
299:input           iRST,   // System Reset(Positive Sync.)
300:input           iCLK    // System Clock
```



```
301:);
302://
303://-----------------------------------------------------------------------------
304:// 水平ライン毎に間引くことによって、垂直方向の削減をする。
305:// 単純にHSyncをゲートすることによって間引く
306://-----------------------------------------------------------------------------
307:  reg           rCamHsi,    rCamHsg;    // Intermediate-HS
308:  reg           rCamVsi;    // Intermediate-VS
309:  reg [15:0]    rCamDti;    // Intermediate-D
310:  reg [ 3:0]    rHsCnt;     // HSyncの立下りでデクリメントするダウン・カウンタ
311:  wire          wCamHssTrg  = &{~iCamHsS,rCamHsi};
312:  wire          wHsGate     = (rHsCnt  == 4'd0);
313:  wire          wHsCntClr   = |{~rCamVsi,iRST};
314:                            // SVO-02:VS立ち上がりとHS立下り同時のため、
315:                            // 一段遅れたrCamVsiを使用
316:  wire          wHsCntSet   = &{wHsGate,wCamHssTrg};
317:  wire[ 4:0]    wHsCntDec   = rHsCnt - 1;
318:  always @( posedge iCLK )
319:  begin
320:    rCamDti     <= iCamDtS;
321:    rCamVsi     <= iCamVsS;
322:    rCamHsi     <= iCamHsS;
323:
324:    if ( iRST )             rCamHsg <= 0;
325:    else                    rCamHsg <= iCamHsS & (wHsGate | ~iENA);
326:
327:    if ( wHsCntClr )        rHsCnt  <= 4'd0;
328:    else if ( wHsCntSet )   rHsCnt  <= iRDCV;           // 水平ラインの数をカウントしながら、
329:    else if ( wCamHssTrg )  rHsCnt  <= wHsCntDec[3:0];  // 垂直方向を間引く
330:    else                    rHsCnt  <= rHsCnt;
331:  end
332:
333:  assign    oCamDtD     = rCamDti;
334:  assign    oCamVsD     = rCamVsi;
335:  assign    oCamHsD     = rCamHsg;  // Gated-HS
336:
337://-----------------------------------------------------------------------------
338:endmodule  // MCamVertReducer
339://-----------------------------------------------------------------------------
340://-----------------------------------------------------------------------------
341:// 水平方向に関する間引きモジュール
342://-----------------------------------------------------------------------------
343:module MCamHorzReducer
344:(
345://-----------------------------------------------------------------------------
346:// General CAM-PORT Source Side
347://
348:input [15:0]    iCamDtS,
349:input           iCamVsS,
350:input           iCamHsS,
351://
352://-----------------------------------------------------------------------------
353:// General CAM-PORT Destination Side
354://
355:output[15:0]    oCamDtD,
356:output          oCamVsD,
357:output          oCamHsD,
358://
359://-----------------------------------------------------------------------------
360:// Control/Status Port
361://
362:input           iENA,       // 前処理機能のイネーブル
363:input [ 3:0]    iRDCH,      // Reduce-H
364:input [ 1:0]    iCPPXL,     // Clock/Pixel -1
365://
366://-----------------------------------------------------------------------------
367:// Common System Lines
368:// カメラのデータ・クロックと同じクロック。
369:input           iRST,   // System Reset(Positive Sync.)
370:input           iCLK    // System Clock
371:);
372://
373://-----------------------------------------------------------------------------
374:// 水平ライン中のピクセルを間引く。最終的な出力として、有効なHSyncアサート中、
375:// 連続な有効画素データを出力する必要があるので、
```



```
376:// ラインバッファを持って、一ラインつめてから、出力するようにする。
377://-----------------------------------------------------------------------------
378:  reg [15:0]    rCamDtd;    assign  oCamDtD = rCamDtd;
379:  reg           rCamVsd;    assign  oCamVsD = rCamVsd;
380:  reg           rCamHsd;    assign  oCamHsD = rCamHsd;
381:  reg [15:0]    rCamDts;
382:  reg           rCamVss;
383:  reg           rCamHss;
384:
385:  // Line Buffer, Width: 16+2(HS/VS), Depth: 4096
386:  wire[17:0]    wFds    = {rCamVss,rCamHss,rCamDts};
387:  wire          wFes;
388:  wire[17:0]    wFdd;
389:  wire          wFed;
390:  MFifoCamLineBuffer
391:    mFifoCamLineBuffer(
392:        .clk(iCLK),     .srst(iRST),
393:        .din(wFds),     .wr_en(wFes),   .full(),
394:        .dout(wFdd),    .rd_en(wFed),   .empty()
395:    );
396:
397://-----------------------------------------------------------------------------
398:  reg           rLnAck;     // Line ACK, バッファに1ライン格納済み
399:  wire          wCamHsd     = wFdd[16];
400:  wire          wCamVsd     = wFdd[17];
401:  assign        wFed        = |{rLnAck,~wCamHsd,~iENA};
402:  wire          wLnAckClr   = &{~wCamHsd,rCamHsd} | iRST;
403:  wire          wLnAckSet   = &{~iCamHsS,rCamHss};
404:  always @( posedge iCLK )
405:  begin
406:    rCamDtd     <= wFdd[15:0];
407:    rCamVsd     <= wCamVsd;
408:    rCamHsd     <= &{wCamHsd,|{rLnAck,~iENA}};
409:    rCamDts     <= iCamDtS;
410:    rCamVss     <= iCamVsS;
411:    rCamHss     <= iCamHsS;
412:
413:    if ( wLnAckClr )        rLnAck  <= 0;
414:    else if ( wLnAckSet )   rLnAck  <= 1;
415:    else                    rLnAck  <= rLnAck;
416:  end
417:
418://-----------------------------------------------------------------------------
419:  reg [ 3:0]    rPiCnt;     // Pixel単位でデクリメントするダウン・カウンタ
420:  reg [ 1:0]    rDiCnt;     // Data単位でデクリメントするダウン・カウンタ
421:  wire          wPiVld      = (rPiCnt == 4'd0);
422:  wire          wDiLim      = (rDiCnt == 2'd0);
423:  wire[ 4:0]    wPiCntDec   = rPiCnt - 1;
424:  wire          wPiCntClr   = &{iCamHsS,~rCamHss};
425:  wire          wPiCntSet   = wPiVld & wDiLim;
426:  wire          wPiCntTrg   = wDiLim;
427:  wire[ 2:0]    wDiCntDec   = rDiCnt - 1;
428:  wire          wDiCntSet   = |{wDiLim,wPiCntClr};
429:
430:  assign        wFes        = |{wPiVld,~iENA};
431:
432:  always @( posedge iCLK )
433:  begin
434:    if ( wPiCntClr )        rPiCnt  <= 4'd0;
435:    else if ( wPiCntSet )   rPiCnt  <= iRDCH;
436:    else if ( wPiCntTrg )   rPiCnt  <= wPiCntDec[3:0];
437:    else                    rPiCnt  <= rPiCnt;
438:    //
439:    if ( wDiCntSet )        rDiCnt  <= iCPPXL;
440:    else                    rDiCnt  <= wDiCntDec[1:0];
441:  end
442:
443://-----------------------------------------------------------------------------
444:endmodule  // MCamHorzReducer
445://-----------------------------------------------------------------------------
```

