LDL07D26A

LDLABO

2007 年 5 月 16 日

目次

図目次

表目次

第 1 章

ビデオディスプレイコントローラ

↪ p.11

1.1 機能

CRT あるいは LCD のモニタのノンインターレースの 640×480 の 256 色のビデオディスプレイコントローラです。VRAM はコントローラ管理で画素データの書き込みもコントローラを経由して行います。

1.1.1 画面

図 1.1 画面構成

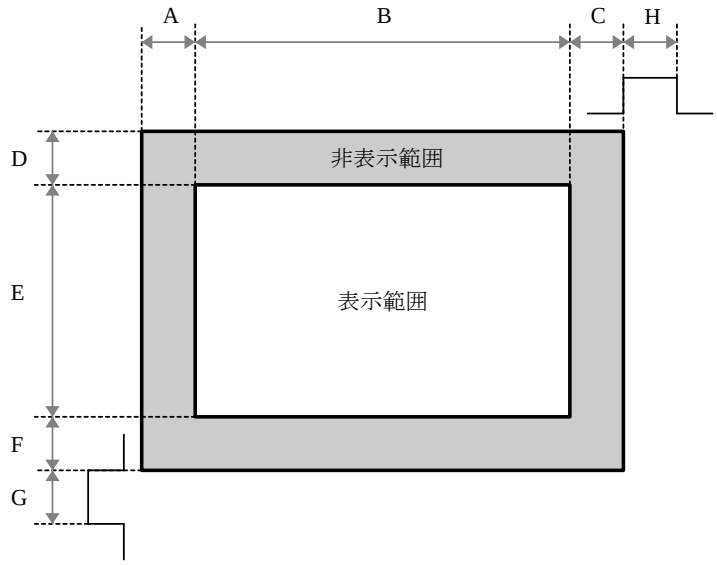


図 1.2 画面定数

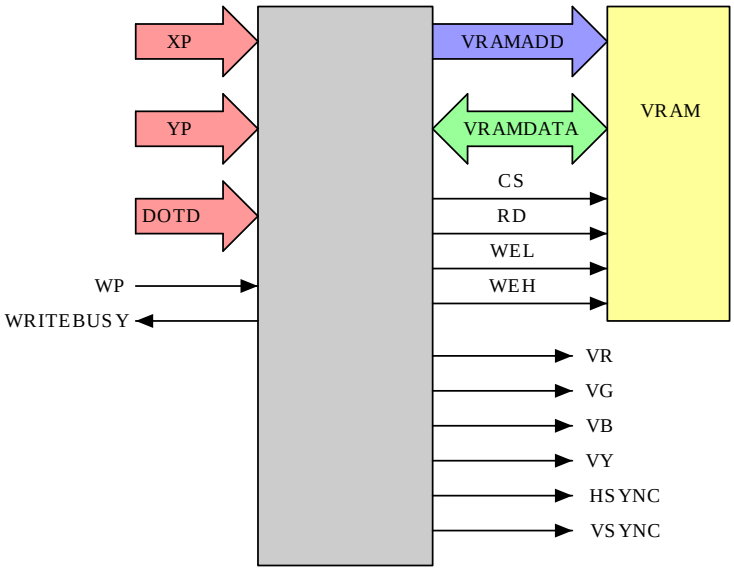
項目	数値	信号
A	48	consthbb
B	640	consthab
C	16	consthsoff
D	33	constvbb
E	480	constvab
F	10	constvsoff
G	2	constvswide
H	96	consthswide

画面構成は図 1.1 のようになっています。右端には水平同期信号、下端には垂直同期信号があります。数値は表 1.2 を見てください。数値は画素の数です。チップ CLK には 24MHz を使います、水平の総画素数は 800 なので 24MHz÷800 で水平周波数は 30KHz になります。垂直の総画素数は 525 なので 30KHz÷525 で垂直周波数は約 57Hz になります。画素周波数は 24MHz です。41.6ns で 1 画素のデータ 8 ビットを送り出します。

1.1.2　VRAM

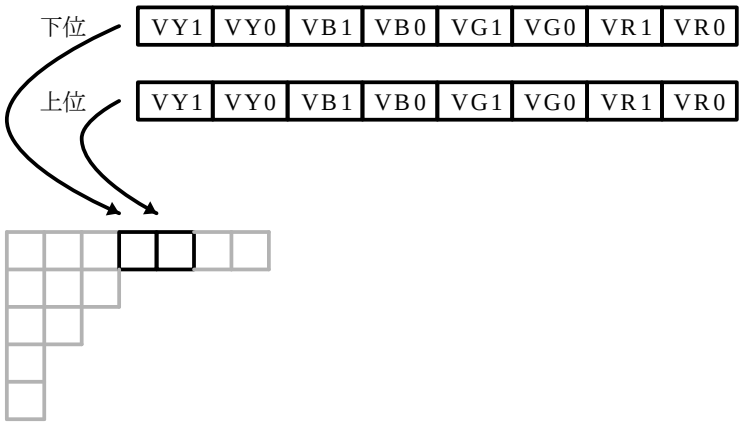
水平画素を 10 ビットで下位、垂直画素を 9 ビットで上位として合わせて 19 ビットの番地でメモリを操作します。1 回のメモリ操作で 2 画素を読むのでデータバスは 16 ビットにします。メモリ操作の周波数は 24MHz÷2 で 12MHz になります。VRAM への書き込みはコントローラを通して行われます。水平位置と垂直位置と画素データを置いて書き込み起点を与えるとコントローラが非表示期間を選んでメモリ操作を行い書き込みが終了すると外部に書き込み終了を知らせます。外部はこれを待って次の書き込みを行います。

図 1.3　構成



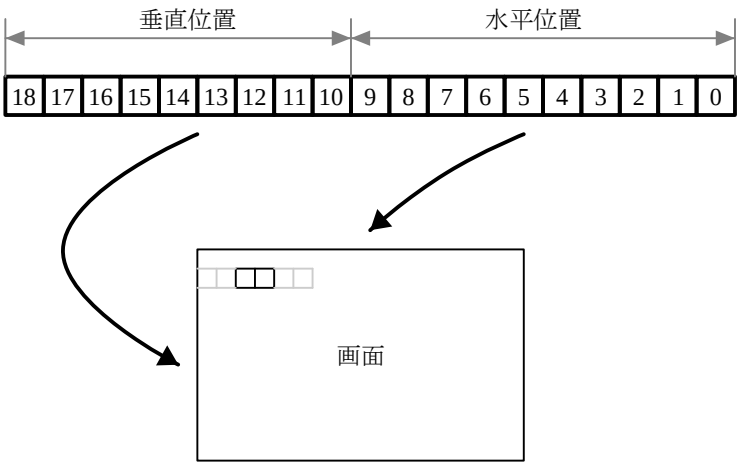
メモリデータの 16 ビットは図 1.4 のように下位が 1 番目の画素、上位が 2 番目の画素と配置されています。

図 1.4　データ画素配置



番地と画素の関係は図 1.5 のようになっています。8 ビット画素で 16 ビットを読むのでひとつの番地で 2 画素を持ちます。水平位置は 0 から 319、垂直位置は 0 から 479 になります。

図 1.5　番地画素配置



1.2　入力

↪ p.11

1.3　出力

↪ p.11

1.4　内部信号

↪ p.11

1.5　検査端子代入

↪ p.12

1.6　出力代入

↪ p.12

1.7　定数設定

↪ p.13

1.8　VRAM 読み出し中

↪ p.13

hst　：水平行程を示します。

1.9　VRAM データ制御

↪ p.13

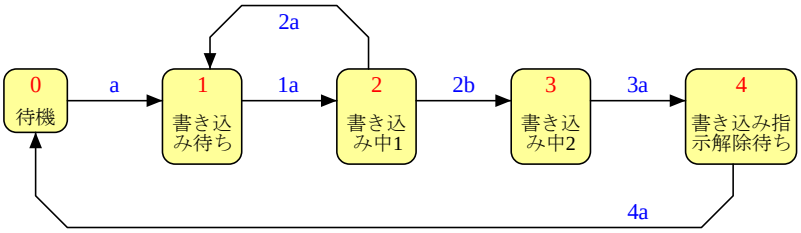
vramread　：VRAM の読み出し中を示します。

1.10　VRAM 書き込み行程

↪ p.13

mwop　：メモリの書き込み行程を示します。
WP　：書き込み起点を示します。
writeprotect　：書き込みの禁止範囲を示します。

図 1.6　VRAM 書き込み行程



1.11　書き込み行程中

↪ p.14

mwop　：メモリの書き込み行程を示します。

1.12　VRAM 書き込み行程

↪ p.14

mwop　：メモリの書き込み行程を示します。

1.13　画素送り

↪ p.14

dotseq　：画素送りの計数を示します。

1.14 VRAM 読み込み 1 番目

↪ p.14

dotseq ：画素送りを示します。
VRAMDATAIN ：VRAM データ入力を示します。

1.15 VRAM 読み込み 2 番目

↪ p.14

dotseq ：画素送りを示します。
VRAMDATAIN ：VRAM データ入力を示します。

1.16 水平表示位置

↪ p.15

hst ：水平行程を示します。
hadd ：水平画素位置を示します。
consthbb ：水平開始非表示を示します。

1.17 水平終了非表示位置

↪ p.15

hst ：水平行程を示します。
hadd ：水平画素位置を示します。
consthab ：水平終了非表示を示します。

1.18 水平同期開始位置

↪ p.15

hst ：水平行程を示します。
hadd ：水平画素位置を示します。
consthsoff ：水平同期開始位置を示します。

1.19 水平同期終了位置

↪ p.15

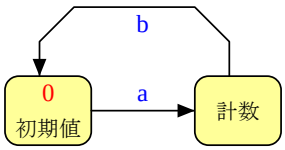
hst　：水平行程を示します。
hadd　：水平画素位置を示します。
consthswide　：垂直同期幅を示します。

1.20　水平画素位置

↪ p.15

RESET　：チップ初期化を示します。
hadd1p　：水平表示位置を示します。
hadd2p　：水平終了非表示位置を示します。
hadd3p　：水平同期開始位置を示します。
hadd4p　：水平同期終了位置を示します。
hadd　：水平画素位置を示します。

図 1.7　水平画素位置



1.21　書き込み禁止範囲

↪ p.15

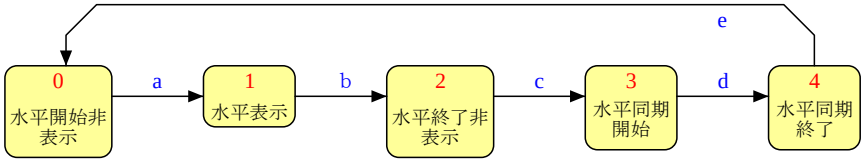
hst　：水平行程を示します。
hadd　：水平画素位置を示します。
constwp　：前非表示の書き込み禁止位置を示します。
vramread　：VRAM の読み出し中を示します。

1.22　水平行程

↪ p.16

RESET　：チップ初期化を示します。
hst　：水平行程を示します。
hadd1p　：水平表示位置を示します。
hadd2p　：水平終了非表示位置を示します。
hadd3p　：水平同期開始位置を示します。
hadd4p　：水平同期終了位置を示します。

図 1.8　水平行程



1.23　垂直表示位置

↪ p.16

vst　：垂直行程を示します。
vadd　：垂直画素位置を示します。
constvbb　：垂直開始非表示を示します。

1.24　垂直終了非表示位置

↪ p.16

vst　：垂直行程を示します。
vadd　：垂直画素位置を示します。
constvab　：垂直終了非表示を示します。

1.25　垂直同期開始位置

↪ p.16

vst　：垂直行程を示します。
vadd　：垂直画素位置を示します。
constvsoff　：垂直同期開始位置を示します。

1.26　垂直同期終了位置

↪ p.16

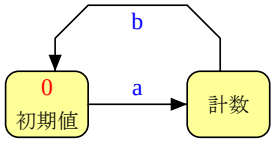
vst　：垂直行程を示します。
vadd　：垂直画素位置を示します。
constvswide　：垂直同期幅を示します。

1.27　垂直画素位置

↪ p.16

RESET　：チップ初期化を示します。
hst　：水平行程を示します。
vadd1p　：垂直表示位置を示します。
vadd2p　：垂直終了非表示位置を示します。
vadd3p　：垂直同期開始位置を示します。
vadd4p　：垂直同期終了位置を示します。

図 1.9　垂直画素位置

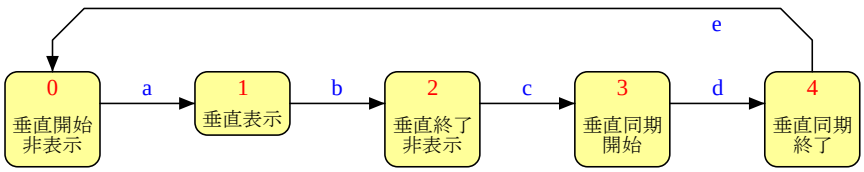


1.28　垂直行程

↪ p.17

RESET　：チップ初期化を示します。
vst　：垂直行程を示します。
vadd1p　：垂直表示位置を示します。
vadd2p　：垂直終了非表示位置を示します。
vadd3p　：垂直同期開始位置を示します。
vadd4p　：垂直同期終了位置を示します。

図 1.10　垂直行程



1.29　機能実行譜

↪ p.18

1.30　端子

↪ p.18

1.31　内部信号

↪ p.18

1.32 実効譜引用

↪ p.18

1.33 検査譜

↪ p.18

第 2 章

論理譜

```
{ ================================================================ }
{   ビデオディスプレイコントローラ                                   }
{ ================================================================ }
logicname LDL07D26A

entity main
{ ---------------------------------------------------------------- }
{   入力                                                           }
{ ---------------------------------------------------------------- }
input  RESET;            { チップ初期化 }
input  WP;               { 書き込み起点 }
input  XP[10];           { 横位置 }
input  YP[9];            { 縦位置 }
input  DOTD[8];          { 画素情報 }
input  VRAMDATAIN[16];   { VRAM データ入力 }

{ ---------------------------------------------------------------- }
{   出力                                                           }
{ ---------------------------------------------------------------- }
output VRAMADD[19];      { VRAM 番地 }
output HSYNC;            { 水平同期 }
output VSYNC;            { 垂直同期 }
output VR[2];            { R 画素 }
output VG[2];            { G 画素 }
output VB[2];            { B 画素 }
output VY[2];            { Y 画素 }
output WRITEBUSY;        { 書き込み中 }
output RD;               { VRAM 読み出し }
output WEL;              { VRAM 下位書き込み }
output WEH;              { VRAM 上位書き込み }
output CS;               { VRAM 選択 }
output VRAMDATAOUT[16];  { VRAM データ出力 }
output VRAMDATACNT;      { VRAM データ制御 }

{ ---------------------------------------------------------------- }
{   内部信号                                                       }
{ ---------------------------------------------------------------- }
bitr   hst[3];           { 水平行程 }
bitr   hadd[10];         { 水平画素位置 }
bitn   consthab[10];     { 水平終了非表示 }
bitn   consthbb[10];     { 水平開始非表示 }
bitn   consthsoff[10];   { 水平同期開始位置 }
bitn   consthswide[10];  { 水平同期幅 }
bitr   vst[3];           { 垂直行程 }
bitr   vadd[9];          { 垂直画素位置 }
```

```
bitn   constvab[9];      { 垂直終了非表示 }
bitn   constvbb[9];      { 垂直開始非表示 }
bitn   constvsoff[9];    { 垂直同期開始位置 }
bitn   constvswide[9];   { 垂直同期幅 }
bitn   hadd1p;           { 水平表示位置 }
bitn   hadd2p;           { 水平終了非表示位置 }
bitn   hadd3p;           { 水平同期開始位置 }
bitn   hadd4p;           { 水平同期終了位置 }
bitn   vadd1p;           { 垂直表示位置 }
bitn   vadd2p;           { 垂直終了非表示位置 }
bitn   vadd3p;           { 垂直同期開始位置 }
bitn   vadd4p;           { 垂直同期終了位置 }
bitr   vdot0r[2];        { 画素 0R }
bitr   vdot1r[2];        { 画素 1R }
bitr   vdot0g[2];        { 画素 0G }
bitr   vdot1g[2];        { 画素 1G }
bitr   vdot0b[2];        { 画素 0B }
bitr   vdot1b[2];        { 画素 1B }
bitr   vdot0y[2];        { 画素 0Y }
bitr   vdot1y[2];        { 画素 1Y }
bitr   dotseq;           { 画素送り }
bitr   mwop[4];          { メモリ書き込み行程 }
bitn   vramread;         { VRAM 読み出し中 }
bitr   vramwrite;        { VRAM 書き込み中 }
bitn   busy;             { 書き込み行程中 }
bitn   writeprotect;     { 書き込み禁止範囲 }
bitn   constwp[10];      { 前非表示の書き込み禁止位置 }

{ ------------------------------------------------------------------ }
{   検査端子代入                                                     }
{ ------------------------------------------------------------------ }

output T0P[3];   T0P=hst;
output T1P[10];  T1P=hadd;
output T2P[3];   T2P=vst;
output T3P[9];   T3P=vadd;
output T4P;      T4P=dotseq;
output T5P;      T5P=writeprotect;


{ ------------------------------------------------------------------ }
{   出力代入                                                         }
{ ------------------------------------------------------------------ }
   if (vramread)
      VRAMADD.0:8=hadd.1:9;
      VRAMADD.9=0;
      VRAMADD.10:18=vadd;
   else
      VRAMADD.0:8=XP.1:9;
      VRAMADD.9=0;
      VRAMADD.10:18=YP;
   endif

   RD=vramread;

   if (!vramread)
      if (vramwrite)
         WEH=XP.0;
         WEL=!XP.0;
      endif
   endif

   if (busy|vramread) CS=1; endif
```

```
   VRAMDATAOUT.0:7=DOTD;
   VRAMDATAOUT.8:15=DOTD;

   WRITEBUSY=busy;

   HSYNC=hst==3;
   VSYNC=vst==3;

   if (dotseq)
      VR.0=vdot0r.1;
      VR.1=vdot1r.1;
      VG.0=vdot0g.1;
      VG.1=vdot1g.1;
      VB.0=vdot0b.1;
      VB.1=vdot1b.1;
      VY.0=vdot0y.1;
      VY.1=vdot1y.1;
   else
      VR.0=vdot0r.0;
      VR.1=vdot1r.0;
      VG.0=vdot0g.0;
      VG.1=vdot1g.0;
      VB.0=vdot0b.0;
      VB.1=vdot1b.0;
      VY.0=vdot0y.0;
      VY.1=vdot1y.0;
   endif

{ ------------------------------------------------------------------- }
{   定数設定                                                          }
{ ------------------------------------------------------------------- }
   consthbb=48;           { 水平開始非表示 }
   consthab=640;          { 水平終了非表示 }
   consthsoff=16;         { 水平同期開始位置 }
   consthswide=96;        { 水平同期幅 }
   constvbb=33;           { 垂直開始非表示 }
   constvab=480;          { 垂直終了非表示 }
   constvsoff=10;         { 垂直同期開始位置 }
   constvswide=2;         { 垂直同期幅 }
   constwp=consthbb-3;    { 前非表示の書き込み禁止位置 }

{ ------------------------------------------------------------------- }
{   VRAM 読み出し中                                                   }
{ ------------------------------------------------------------------- }
   vramread=hst==1;

{ ------------------------------------------------------------------- }
{   VRAM データ制御                                                   }
{ ------------------------------------------------------------------- }
   VRAMDATACNT=!vramread;

{ ------------------------------------------------------------------- }
{   VRAM 書き込み行程                                                 }
{ ------------------------------------------------------------------- }
   if (RESET)
      mwop=0;
   else
      switch(mwop)
         case 0:                    { 待機 }
            if (WP) mwop=1; endif   { a }
         case 1:                    { 書き込み待ち }
            if (writeprotect)
               mwop=mwop;
```

```
            else
               mwop=2;              { 1a }
            endif
         case 2:                    { 書き込み中 1 }
            if (writeprotect)
               mwop=1;              { 2a }
            else
               mwop=3;              { 2b }
            endif
         case 3:                    { 書き込み中 2 }
            mwop=4;                 { 3a }
         case 4:                    { 書き込み指示解除待ち }
            if (!WP)
               mwop=0;              { 4a }
            else
               mwop=mwop;
            endif
      endswitch
   endif

{ ------------------------------------------------------------------ }
{   書き込み行程中                                    }
{ ------------------------------------------------------------------ }
   if (mwop>0) busy=1; endif

{ ------------------------------------------------------------------ }
{   VRAM 書き込み行程                                 }
{ ------------------------------------------------------------------ }
   if ((mwop==2)|(mwop==3)) vramwrite=1 ; endif

{ ------------------------------------------------------------------ }
{   画素送り                                          }
{ ------------------------------------------------------------------ }
   if (RESET)
      dotseq=0;
   else
      dotseq=dotseq+1;
   endif

{ ------------------------------------------------------------------ }
{   VRAM 読み込み１番目                               }
{ ------------------------------------------------------------------ }
   if (dotseq)
      vdot0r.0=VRAMDATAIN.0;
      vdot1r.0=VRAMDATAIN.1;
      vdot0g.0=VRAMDATAIN.2;
      vdot1g.0=VRAMDATAIN.3;
      vdot0b.0=VRAMDATAIN.4;
      vdot1b.0=VRAMDATAIN.5;
      vdot0y.0=VRAMDATAIN.6;
      vdot1y.0=VRAMDATAIN.7;
   else
      vdot0r.0=vdot0r.0;
      vdot1r.0=vdot1r.0;
      vdot0g.0=vdot0g.0;
      vdot1g.0=vdot1g.0;
      vdot0b.0=vdot0b.0;
      vdot1b.0=vdot1b.0;
      vdot0y.0=vdot0y.0;
      vdot1y.0=vdot1y.0;
   endif

{ ------------------------------------------------------------------ }
{   VRAM 読み込み２番目                               }
{ ------------------------------------------------------------------ }
```

```
   if (dotseq)
      vdot0r.1=VRAMDATAIN.8;
      vdot1r.1=VRAMDATAIN.9;
      vdot0g.1=VRAMDATAIN.10;
      vdot1g.1=VRAMDATAIN.11;
      vdot0b.1=VRAMDATAIN.12;
      vdot1b.1=VRAMDATAIN.13;
      vdot0y.1=VRAMDATAIN.14;
      vdot1y.1=VRAMDATAIN.15;
   else
      vdot0r.1=vdot0r.1;
      vdot1r.1=vdot1r.1;
      vdot0g.1=vdot0g.1;
      vdot1g.1=vdot1g.1;
      vdot0b.1=vdot0b.1;
      vdot1b.1=vdot1b.1;
      vdot0y.1=vdot0y.1;
      vdot1y.1=vdot1y.1;
   endif

{ ------------------------------------------------------------------ }
{   水平表示位置                                    }
{ ------------------------------------------------------------------ }
   if (hst==0)
      if (hadd==consthbb) hadd1p=1; endif
   endif

{ ------------------------------------------------------------------ }
{   水平終了非表示位置                                }
{ ------------------------------------------------------------------ }
   if (hst==1)
      if (hadd==consthab) hadd2p=1; endif
   endif

{ ------------------------------------------------------------------ }
{   水平同期開始位置                                  }
{ ------------------------------------------------------------------ }
   if (hst==2)
      if (hadd==consthsoff) hadd3p=1; endif
   endif

{ ------------------------------------------------------------------ }
{   水平同期終了位置                                  }
{ ------------------------------------------------------------------ }
   if (hst==3)
      if (hadd==consthswide) hadd4p=1; endif
   endif

{ ------------------------------------------------------------------ }
{   水平画素位置                                    }
{ ------------------------------------------------------------------ }
   if (RESET)
      hadd=0;                              { b }
   else
      if (hadd1p|hadd2p|hadd3p|hadd4p)
         hadd=0;                           { b }
      else
         hadd=hadd+1;                      { a }
      endif
   endif

{ ------------------------------------------------------------------ }
{   書き込み禁止範囲                                  }
{ ------------------------------------------------------------------ }
   if (hst==0)
```

```
     if (hadd>constwp) writeprotect=1; endif
   else
      if (vramread) writeprotect=1; endif
   endif

{ ------------------------------------------------------------------ }
{   水平行程                                          }
{ ------------------------------------------------------------------ }
   if (RESET)
      hst=0;                                              { e }
   else
      switch(hst)
         case 0:                                          { 水平開始非表示 }
            if (hadd1p) hst=1; endif                      { a }
         case 1:                                          { 水平表示 }
            if (hadd2p) hst=2; else hst=hst; endif        { b }
         case 2:                                          { 水平終了非表示 }
            if (hadd3p) hst=3; else hst=hst; endif        { c }
         case 3:                                          { 水平同期開始 }
            if (hadd4p) hst=4; else hst=hst; endif        { d }
         case 4:                                          { 水平同期終了 }
            hst=0;                                        { e }
      endswitch
   endif

{ ------------------------------------------------------------------ }
{   垂直表示位置                                      }
{ ------------------------------------------------------------------ }
   if (vst==0)
      if (vadd==constvbb) vadd1p=1; endif
   endif

{ ------------------------------------------------------------------ }
{   垂直終了非表示位置                                }
{ ------------------------------------------------------------------ }
   if (vst==1)
      if (vadd==constvab) vadd2p=1; endif
   endif

{ ------------------------------------------------------------------ }
{   垂直同期開始位置                                  }
{ ------------------------------------------------------------------ }
   if (vst==2)
      if (vadd==constvsoff) vadd3p=1; endif
   endif

{ ------------------------------------------------------------------ }
{   垂直同期終了位置                                  }
{ ------------------------------------------------------------------ }
   if (vst==3)
      if (vadd==constvswide) vadd4p=1; endif
   endif

{ ------------------------------------------------------------------ }
{   垂直画素位置                                      }
{ ------------------------------------------------------------------ }
   if (RESET)
      vadd=0;                                  { b }
   else
      if (vadd1p|vadd2p|vadd3p|vadd4p)
         vadd=0;                               { b }
      else
         if (hst==4)
            vadd=vadd+1;                       { a }
```

```
         else
            vadd=vadd;
         endif
      endif
   endif

{ ------------------------------------------------------------------ }
{   垂直行程                                          }
{ ------------------------------------------------------------------ }
   if (RESET)
      vst=0;                                              { e }
   else
      switch(vst)
         case 0:                                          { 垂直開始非表示 }
            if (vadd1p) vst=1; else vst=vst; endif        { a }
         case 1:                                          { 垂直表示 }
            if (vadd2p) vst=2; else vst=vst; endif        { b }
         case 2:                                          { 垂直終了非表示 }
            if (vadd3p) vst=3; else vst=vst; endif        { c }
         case 3:                                          { 垂直同期開始 }
            if (vadd4p) vst=4; else vst=vst; endif        { d }
         case 4:                                          { 垂直同期終了 }
            vst=0;                                        { e }
      endswitch
   endif

ende
```

```
{ ================================================================ }
{    機能実行譜                                                    }
{ ================================================================ }
entity sim
{ ---------------------------------------------------------------- }
{    端子                                                          }
{ ---------------------------------------------------------------- }
output RESET;
output WP;
output XP[10];
output YP[9];
output DOTD[8];
output VRAMDATAIN[16];
output VRAMADD[19];
output HSYNC;
output VSYNC;
output VR[2];
output VG[2];
output VB[2];
output VY[2];
output WRITEBUSY;
output RD;
output WEL;
output WEH;
output CS;
output VRAMDATAOUT[16];
output VRAMDATACNT;
input  simres;

{ ---------------------------------------------------------------- }
{    内部信号                                                      }
{ ---------------------------------------------------------------- }
bitr tc[20];
bitn node_VRAMADD[19];

{ ---------------------------------------------------------------- }
{    内部信号閲覧                                                  }
{ ---------------------------------------------------------------- }
output TP[20]; TP=tc;                    { 検査位置 }

{ ---------------------------------------------------------------- }
{    実効譜引用                                                    }
{ ---------------------------------------------------------------- }
   part main(RESET,WP,XP,YP,DOTD,VRAMDATAIN,node_VRAMADD,HSYNC,VSYNC
            ,VR,VG,VB,VY,WRITEBUSY,RD,WEL,WEH,CS,VRAMDATAOUT,VRAMDATACNT)

   VRAMADD=node_VRAMADD;

{ ---------------------------------------------------------------- }
{    検査譜                                                        }
{ ---------------------------------------------------------------- }
   simres=0;
   if (!simres) tc=tc+1; endif

   if (tc<5) RESET=1; endif

  XP=0x12;
  YP=0x34;
  DOTD=0x55;

   switch(node_VRAMADD)
      case 1: VRAMDATAIN=0x5566;
      case 2: VRAMDATAIN=0x7788;
```

```
   endswitch

   if (tc==7) WP=1; endif
   if (tc==83) WP=1; endif

ende

endlogic
```