LDL08A24A

LDLABO

2008 年 6 月 22 日

第 1 章

論理譜

```
{ ========================================================================== }
{    データ分離器                                                          }
{ ========================================================================== }
logicname LDL08A24A

{ ========================================================================== }
{    実効譜                                                                }
{ ========================================================================== }
entity main

{ -------------------------------------------------------------------------- }
{    入力                                                                  }
{ -------------------------------------------------------------------------- }
input RESET;          { 初期化 }
input RELOCK;         { FDCR : 再同期開始 }
input RDATA;          { FDD : Disk 読み出しデータ }
input MFM;            { CPU : 単密/倍密切り換え }

{ -------------------------------------------------------------------------- }
{    出力                                                                  }
{ -------------------------------------------------------------------------- }
output Q[2];

bitn   WINDOW;        { データ窓 }
bitn   LOCK;          { 同期確立 }

{ -------------------------------------------------------------------------- }
{    内部信号                                                              }
{ -------------------------------------------------------------------------- }
bitr  window[2];      { データ窓 }
bitr  lock;           { 同期確立 }

bitr  rpulse[2];      { 元信号パルス }
bitr  pc[8];          { 周期計数 }
bitr  ps[8];          { 周期設定 }
bitr  wct[8];         { 窓計数 }
bitr  lockct[8];      { 同期検出計数 }
bitn  loadw0;         { 周期代入パルス }
bitr  ss[8];          { 同期後周期設定 }
bitn  sseq;           { 窓中心一致比較 }
bitn  scp;            { 窓中心大小比較 }
bitr  lz;             { 左微調整パルス }
bitn  rz;             { 右微調整パルス }
bitr  rzc[8];         { 増調整パルス幅計数 }
bitr  rzmes;          { 増調整計測 }
bitr  rzdet;          { 増調整発効 }
```

```
bitn  sslimit;        { 周期下限設定 }
bitn  sel[8];         { 周期設定選択 }
bitr  lockd[4];       { 同期検出 }
bitn  syncout;        { 同期解除 }
bitn  fmsout;         { 単密同期除外 }
bitn  ss4scale[8];    { 同期後周期設定の 4 倍 }
bitn  rdata;          { Disk 読み出しデータ }
bitn  Relock;         { 再同期開始 }
bitn  mfm;            { 単密/倍密切り換え }
bitn  Reset;          { 初期化 }

{ ------------------------------------------------------------------------ }
{    検査端子                                                            }
{ ------------------------------------------------------------------------ }
output T0P[8];  T0P=pc;
output T1P[8];  T1P=ps;
output T2P[8];  T2P=wct;
output T3P;     T3P=lock;
output T4P;     T4P=rz;
output T5P;     T5P=rpulse.0;
output T6P;     T6P=loadw0;
output T7P[8];  T7P=lockct;
output T8P[4];  T8P=0;
output T9P;     T9P=0;
output T10P;    T10P=0;
output T11P[8]; T11P=ss;
output T12P;    T12P=sseq;
output T13P;    T13P=scp;
output T14P;    T14P=lz;
output T15P[8]; T15P=rzc;
output T16P;    T16P=rzmes;
output T17P;    T17P=rzdet;
output T18P;    T18P=sslimit;
output T19P[8]; T19P=sel;
output T20P[4]; T20P=lockd;
output T21P;    T21P=syncout;
output T22P;    T22P=fmsout;
output T23P[8]; T23P=ss4scale;
output T24P;    T24P=rdata;
output T25P;    T25P=Relock;
output T26P;    T26P=mfm;
output T27P;    T27P=Reset;
output T28P[2]; T28P=window;

{ ------------------------------------------------------------------------ }
{    出力代入                                                            }
{ ------------------------------------------------------------------------ }
   Q.0=WINDOW;
   Q.1=LOCK;

   WINDOW=window.0;
   LOCK=lock;

{ ------------------------------------------------------------------------ }
{    入力代入                                                            }
{ ------------------------------------------------------------------------ }
   Reset=RESET;
   rdata=RDATA;
   Relock=RELOCK;
   mfm=MFM;

{ ------------------------------------------------------------------------ }
{    元信号パルス                                                        }
{ ------------------------------------------------------------------------ }
```

```
   if (Reset)
      rpulse=0;
   else
      switch(rpulse)
         case 0: if (rdata) rpulse=1; endif
         case 1: rpulse=2;
         case 2:
           if (rdata)
               rpulse=rpulse;
            else
               rpulse=0;
            endif
      endswitch
   endif

{ ------------------------------------------------------------------------ }
{    周期計数                                                            }
{ ------------------------------------------------------------------------ }
   if (Reset)
      pc=0;
   else
      if (rpulse.0)
         pc=0;
      else
         pc=pc+1;
      endif
   endif

{ ------------------------------------------------------------------------ }
{    周期設定                                                            }
{ ------------------------------------------------------------------------ }
   if (Reset)
      ps=0;
   else
      if (rpulse.0)
         ps=pc;
      else
         ps=ps;
      endif
   endif

{ ------------------------------------------------------------------------ }
{    同期検出                                                            }
{ ------------------------------------------------------------------------ }
   if (!Reset&!Relock)
      switch(lockd)
         case 0:                          { クロックパルス待ち }
            switch(rpulse.0,window.0)
               case 1,0: lockd=1;         { クロックパルス 1 }
            endswitch
         case 1:                          { クロックパルス通過待ち }
            if (RDATA)
               lockd=lockd;
            else
               lockd=2;                   { クロックパルス 0 }
            endif
         case 2:                          { データ窓待ち }
            if (window.0)
               lockd=4;                   { データ窓移行 }
            else
               if (RDATA)
                  lockd=3;                { 同期はずれ }
               else
                  lockd=lockd;
               endif
```

```
            endif
         case 3:                          { 同期外検出 }
            if (window.0)
               lockd=0;                   { 初期化 }
            else
               lockd=lockd;               { データ窓待ち }
            endif
         case 4:                          { データ窓通過待ち }
            if (window.0)
               if (RDATA)
                  lockd=3;                { 同期はずれ }
               else
                  lockd=lockd;
               endif
            else
               lockd=8;                   { データ窓終了 }
            endif
         case 8:                          { ビットセル終了 }
            lockd=0;                      { 初期化 }
      endswitch
   endif

{ ------------------------------------------------------------------------ }
{    同期検出計数                                                          }
{ ------------------------------------------------------------------------ }
   if (!Reset&!Relock&!syncout)
      if (lockd.3)
         lockct=lockct+1;
      else
         lockct=lockct;
      endif
   endif

{ ------------------------------------------------------------------------ }
{    同期確立                                                              }
{ ------------------------------------------------------------------------ }
   if (!Reset&!Relock)
      switch(lock)
         case 0:
            switch(lockct)
               case 16: lock=1;
            endswitch
         case 1:
            lock=lock;
      endswitch
   endif

{ ------------------------------------------------------------------------ }
{    周期設定選択                                                          }
{ ------------------------------------------------------------------------ }
   if (lock)
      sel=ss;
   else
      sel=ps;
   endif

{ ------------------------------------------------------------------------ }
{    窓計数                                                                }
{ ------------------------------------------------------------------------ }
   if (Reset|Relock)
      wct=0;
   else
      if (lock)
         if (rz)
            wct=wct;
```

```
         else
            if (loadw0)
               wct.0:6=sel.1:7;
            else
               wct=wct-1;
            endif
         endif
      else
         switch(rpulse.0,loadw0)
            case 0,0: wct=wct-1;
            case 1,1: wct.0:5=sel.2:7;
            case 0,1: wct.0:6=sel.1:7;
            case 1,0: wct.0:5=sel.2:7;
         endswitch
      endif
   endif

{ ------------------------------------------------------------------------ }
{    周期代入パルス                                                        }
{ ------------------------------------------------------------------------ }
   switch(wct)
      case 0: loadw0=1;
   endswitch

{ ------------------------------------------------------------------------ }
{    データ窓                                                              }
{ ------------------------------------------------------------------------ }
   if (Reset)
      window=0;
   else
      if (lock)
         switch(window)
            case 0: if (loadw0) window=2; endif
            case 2:
               if (loadw0)
                  window=window;
               else
                  window=3;
               endif
            case 3:
               if (loadw0)
                  window=0;
               else
                  window=window;
               endif
         endswitch
      else
         switch(window)
            case 0: if (loadw0) window=1; endif
            case 1:
               switch(loadw0,rpulse.0)
                  case 0,0: window=window;
                  default:  window=0;
               endswitch
         endswitch
      endif
   endif

{ ------------------------------------------------------------------------ }
{    同期後周期設定                                                        }
{ ------------------------------------------------------------------------ }
   if (lock)
      if (sslimit)
         ss=8;
      else
```

```
            switch(rz,lz)
               case 1,0:
                  switch(ss)
                     case 0xff: ss=ss;
                     default:   ss=ss+1;
                  endswitch
               case 0,1:
                  switch(ss)
                     case 0:  ss=ss;
                     default: ss=ss-1;
                  endswitch
               default: ss=ss;
            endswitch
      endif
   else
      ss=ps;
   endif

{ ------------------------------------------------------------------------ }
{    周期下限設定                                                          }
{ ------------------------------------------------------------------------ }
   switch(ss)
      case 0:
      case 1:
      case 2:
      case 3:
      case 4:
      case 5:
      case 6:
      case 7:
      from: sslimit=1;
   endswitch

{ ------------------------------------------------------------------------ }
{    同期後周期設定の 4 倍                                                 }
{ ------------------------------------------------------------------------ }
   ss4scale.6:7=0;
   ss4scale.0:5=ss.2:7;

{ ------------------------------------------------------------------------ }
{    窓中心一致比較                                                        }
{ ------------------------------------------------------------------------ }
   if (wct==ss4scale) sseq=1; endif

{ ------------------------------------------------------------------------ }
{    窓中心大小比較                                                        }
{ ------------------------------------------------------------------------ }
   if (wct>ss4scale) scp=1; endif

{ ------------------------------------------------------------------------ }
{    左微調整パルス                                                        }
{ ------------------------------------------------------------------------ }
   switch(lz)
      case 0: if (!sseq&scp&rpulse.0) lz=1; endif
      case 1:
         switch(sseq,scp)
            case 0,1: lz=lz;
            default : lz=0;
         endswitch
   endswitch

{ ------------------------------------------------------------------------ }
{    増調整計測                                                            }
{ ------------------------------------------------------------------------ }
   if (scp)
```

```
      rzmes=0;
   else
      switch(rzmes)
         case 0: if (sseq) rzmes=1; endif
         case 1: rzmes=rzmes;
      endswitch
   endif

{ ------------------------------------------------------------------------ }
{    増調整発効                                                            }
{ ------------------------------------------------------------------------ }
   if (sseq)
      rzdet=0;
   else
      switch(rzdet)
         case 0:
            switch(rpulse.0,scp)
               case 1,0: rzdet=1;
            endswitch
         case 1:
            rzdet=rzdet;
      endswitch
   endif

{ ------------------------------------------------------------------------ }
{    増調整パルス幅計数                                                    }
{ ------------------------------------------------------------------------ }
   if (Reset|sseq)
      rzc=0;
   else
      switch(rzmes,rzdet)
         case 1,0:
            switch(rzc)
               case 0xff: rzc=rzc;
               default:   rzc=rzc+1;
            endswitch
         default:
            switch(rzc)
               case 0:  rzc=rzc;
               default: rzc=rzc-1;
            endswitch
      endswitch
   endif

{ ------------------------------------------------------------------------ }
{    右微調整パルス                                                        }
{ ------------------------------------------------------------------------ }
   if (rzdet)
      switch(rzc)
         case 0:  rz=0;
         default: rz=1;
      endswitch
   endif

{ ------------------------------------------------------------------------ }
{    単密同期除外                                                          }
{ ------------------------------------------------------------------------ }
   if (ps<=48) fmsout=1; endif

{ ------------------------------------------------------------------------ }
{    同期解除                                                              }
{ ------------------------------------------------------------------------ }
   if (lock)
   else
     if (mfm)
```

```
            switch(lockd)
               case 3: syncout=1;
            endswitch
         else
            syncout=fmsout;
         endif
      endif

ende
```

```
{ ========================================================================== }
{    機能実行譜                                                            }
{ ========================================================================== }
entity sim
output RESET;
output RELOCK;
output RDATA;
output MFM;
output Q[2];

{ -------------------------------------------------------------------------- }
{    検証論理端子                                                          }
{ -------------------------------------------------------------------------- }
output T0BP[3];
output T1BP[16];
output T2BP[7];
output T3BP[7];
output T4BP;
output T5BP[3];
output T6BP[8];

{ -------------------------------------------------------------------------- }
{    内部信号                                                              }
{ -------------------------------------------------------------------------- }
bitr tc[16];          { 検証計数 }
bitr sel[7];          { ビットセル行程 }
bitr wide[7];         { ビットセル幅 }
bitn pulsewide[7];    { パルス幅 }
bitn datapulses[7];   { データパルス開始位置 }
bitn datapulsee[7];   { データパルス終了位置 }
bitn widecenter[7];   { ビットセル中心 }
bitr bitcount[3];     { ビット計数 }
bitr bytecount[16];   { バイト計数 }
bitn datapulse;       { データパルス }
bitn clockpulse;      { クロックパルス }
bitr bitcount[3];     { ビット計数 }
bitn selzero;         { ビットセル開始位置 }
bitn bytedata[8];     { バイトデータ }
bitn bitdata;         { ビットデータ }
bitn byteclock[8];    { MFM クロックデータ }
bitr lastbit;         { MFM データ生成用最終ビット }
bitn bitclock;        { MFM クロックパルス }

bitn node_RESET;      { 初期化 }
bitn node_RELOCK;     { 再同期開始 }
bitn node_MFM;        { 方式選択 }

{ -------------------------------------------------------------------------- }
{    検証論理端子代入                                                      }
{ -------------------------------------------------------------------------- }
   T0BP=bitcount;
   T1BP=bytecount;
   T2BP=datapulses;
   T3BP=datapulsee;
   T4BP=datapulse;
   T5BP=bitcount;
   T6BP=byteclock;

{ -------------------------------------------------------------------------- }
{    方式選択                                                              }
{ -------------------------------------------------------------------------- }
   node_MFM=0;
```

```
{ ------------------------------------------------------------------------ }
{    実効譜引用                                                          }
{ ------------------------------------------------------------------------ }
   part main(node_RESET,RELOCK,RDATA,MFM,Q)

{ ------------------------------------------------------------------------ }
{    検証計数                                                            }
{ ------------------------------------------------------------------------ }
   tc=tc+1;

{ ------------------------------------------------------------------------ }
{    初期化                                                              }
{ ------------------------------------------------------------------------ }
   if (tc<5) node_RESET=1; endif

{ ------------------------------------------------------------------------ }
{    ビットセル行程                                                      }
{ ------------------------------------------------------------------------ }
   if (node_RESET)
      sel=0;
   else
      if (sel>=wide)
         sel=0;
      else
         sel=sel+1;
      endif
   endif

{ ------------------------------------------------------------------------ }
{    クロックパルス                                                      }
{ ------------------------------------------------------------------------ }
   if ((sel<pulsewide)&(sel>0))
      if (node_MFM)
         clockpulse=bitclock;
      else
         clockpulse=1;
      endif
   endif

{ ------------------------------------------------------------------------ }
{    ビットセル開始位置                                                  }
{ ------------------------------------------------------------------------ }
   switch(sel)
      case 0: selzero=1;
   endswitch

{ ------------------------------------------------------------------------ }
{    ビット計数                                                          }
{ ------------------------------------------------------------------------ }
   if (node_RESET)
      bitcount=0;
   else
      if (selzero)
         switch(bitcount)
            case 7:  bitcount=0;
            default: bitcount=bitcount+1;
         endswitch
      else
         bitcount=bitcount;
      endif
   endif

{ ------------------------------------------------------------------------ }
```

```
{    バイト計数                                          }
{ ------------------------------------------------------------------------ }
   if (node_RESET)
      bytecount=0;
   else
      switch(bitcount)
         case 7:
            switch(sel)
               case 0:  bytecount=bytecount+1;
               default: bytecount=bytecount;
            endswitch
         default: bytecount=bytecount;
      endswitch
   endif

{ ------------------------------------------------------------------------ }
{    MFM データ生成用最終ビット                          }
{ ------------------------------------------------------------------------ }
   if (node_RESET)
      lastbit=0;
   else
      switch(bitcount)
         case 7:
            switch(sel)
               case 0:  lastbit=bitdata;
               default: lastbit=lastbit;
            endswitch
         default: lastbit=lastbit;
      endswitch
   endif

{ ------------------------------------------------------------------------ }
{    パルス幅                                            }
{ ------------------------------------------------------------------------ }
   pulsewide.0:3=wide.3:6;

{ ------------------------------------------------------------------------ }
{    データパルス開始位置                                }
{ ------------------------------------------------------------------------ }
   datapulses=widecenter;

{ ------------------------------------------------------------------------ }
{    データパルス終了位置                                }
{ ------------------------------------------------------------------------ }
   datapulsee=widecenter+pulsewide;

{ ------------------------------------------------------------------------ }
{    ビットセル中心                                      }
{ ------------------------------------------------------------------------ }
   widecenter.0:5=wide.1:6;

{ ------------------------------------------------------------------------ }
{    ビットセル幅                                        }
{ ------------------------------------------------------------------------ }
   if (node_MFM)
      switch(bytecount)
         case 5:
            switch(bitcount)
               case 0: wide=30;
               case 1: wide=29;
               case 2: wide=28;
               case 3: wide=26;
               case 4: wide=27;
               case 5: wide=28;
               case 6: wide=29;
```

```
                case 7: wide=30;
             endswitch
          default: wide=31;
       endswitch
    else
       switch(bytecount)
          case 5:
             switch(bitcount)
                case 0: wide=61;
                case 1: wide=59;
                case 2: wide=57;
                case 3: wide=55;
                case 4: wide=57;
                case 5: wide=59;
                case 6: wide=61;
                case 7: wide=63;
             endswitch
          default: wide=63;
       endswitch
    endif

{ ------------------------------------------------------------------------ }
{    データパルス                                                          }
{ ------------------------------------------------------------------------ }
    if ((datapulses<=sel)&(datapulsee>=sel))
       if (bitdata)
          datapulse=1;
       endif
    endif

{ ------------------------------------------------------------------------ }
{    バイトデータ                                                          }
{ ------------------------------------------------------------------------ }
    if (node_MFM)
       switch(bytecount)
          case 0:  bytedata=0x4e;   { GAP }
          case 1:  bytedata=0x4e;   { GAP }
          case 2:  bytedata=0;      { SYNC }
          case 3:  bytedata=0;      { SYNC }
          case 4:  bytedata=0;      { SYNC }
          case 5:  bytedata=0;      { SYNC }
          case 6:  bytedata=0;      { SYNC }
          case 7:  bytedata=0;      { SYNC }
          default: bytedata=0x55;
       endswitch
    else
       switch(bytecount)
          case 0:  bytedata=0xff;   { GAP }
          case 1:  bytedata=0xff;   { GAP }
          case 2:  bytedata=0;      { SYNC }
          case 3:  bytedata=0;      { SYNC }
          case 4:  bytedata=0;      { SYNC }
          case 5:  bytedata=0;      { SYNC }
          case 6:  bytedata=0;      { SYNC }
          case 7:  bytedata=0;      { SYNC }
          default: bytedata=0x55;
       endswitch
    endif

{ ------------------------------------------------------------------------ }
{    ビットデータ                                                          }
{ ------------------------------------------------------------------------ }
    switch(bitcount)
       case 0: bitdata=bytedata.0;
```

```
      case 1: bitdata=bytedata.1;
      case 2: bitdata=bytedata.2;
      case 3: bitdata=bytedata.3;
      case 4: bitdata=bytedata.4;
      case 5: bitdata=bytedata.5;
      case 6: bitdata=bytedata.6;
      case 7: bitdata=bytedata.7;
   endswitch

{ ------------------------------------------------------------------- }
{    MFM クロックデータ                                              }
{ ------------------------------------------------------------------- }
   if (lastbit|bytedata.0) else byteclock.0=1; endif
   if (bytedata.0|bytedata.1) else byteclock.1=1; endif
   if (bytedata.1|bytedata.2) else byteclock.2=1; endif
   if (bytedata.2|bytedata.3) else byteclock.3=1; endif
   if (bytedata.3|bytedata.4) else byteclock.4=1; endif
   if (bytedata.4|bytedata.5) else byteclock.5=1; endif
   if (bytedata.5|bytedata.6) else byteclock.6=1; endif
   if (bytedata.6|bytedata.7) else byteclock.7=1; endif

{ ------------------------------------------------------------------- }
{    MFM クロックパルス                                              }
{ ------------------------------------------------------------------- }
   switch(bitcount)
      case 0: bitclock=byteclock.0;
      case 1: bitclock=byteclock.1;
      case 2: bitclock=byteclock.2;
      case 3: bitclock=byteclock.3;
      case 4: bitclock=byteclock.4;
      case 5: bitclock=byteclock.5;
      case 6: bitclock=byteclock.6;
      case 7: bitclock=byteclock.7;
   endswitch

{ ------------------------------------------------------------------- }
{    Disk 読み出しデータ                                             }
{ ------------------------------------------------------------------- }
   RDATA=clockpulse|datapulse;

{ ------------------------------------------------------------------- }
{    初期化                                                          }
{ ------------------------------------------------------------------- }
   RESET=node_RESET;

{ ------------------------------------------------------------------- }
{    再同期開始                                                      }
{ ------------------------------------------------------------------- }
   RELOCK=node_RELOCK;

{ ------------------------------------------------------------------- }
{    単密/倍密切り換え                                               }
{ ------------------------------------------------------------------- }
   MFM=node_MFM;

ende

endlogic
```