

LDL08E12A

LDLABO

2008年6月14日



# 第 1 章

## 論理譜

```

{ ===== }
{   BCD 変換器   }
{ ===== }
logicname LDL08E12A

{ ===== }
{   実効譜   }
{ ===== }
entity main
{ ----- }
{   入力   }
{ ----- }
input  RESET;      { 初期化 }
input  REQ;        { 変換要求 }
input  ADDATA[8];  { データ }

{ ----- }
{   出力   }
{ ----- }
output BCD0P[4];   { BCD 第 1 桁 }
output BCD1P[4];   { BCD 第 2 桁 }
output BCD2P[4];   { BCD 第 3 桁 }
output ACK;        { 変換完了 }

{ ----- }
{   内部信号   }
{ ----- }
bitn  addata[16];  { 変換前データ }
bitn  noa[8];      { 計算 1 }
bitn  nob[8];      { 計算値 3 代入 }
bitn  noc[8];      { 計算値 4 代入 }
bitn  nod[16];     { 計算 2 }
bitn  noe[16];     { 計算値 1 代入 }
bitn  nof[16];     { 計算値 2 代入 }

bitr  sequence[2]; { 行程 }
bitr  delaysequence[2]; { 遅延行程 }
bitn  divres;      { 割り算初期化 }
bitn  eoc;         { 計算 1 終了位置 }
bitn  eocb;        { 計算 2 終了位置 }
bitr  deg2p[4];    { BCD3 桁 }
bitr  deg1p[4];    { BCD2 桁 }
bitr  deg0p[4];    { BCD1 桁 }
bitr  req[2];      { 要求行程 }
bitn  resseq;      { 行程初期化 }
bitn  sequence2p;  { 行程 2 位置 }

```

```

bitn sequence3p;      { 行程 3 位置 }
bitr ack[2];         { 変換完了行程 }
bitn ireq;           { 変換要求 }

{ ----- }
{  検査端子代入      }
{ ----- }

output TOP[16]; TOP=addata;
output T1P[8]; T1P=noa;
output T2P[8]; T2P=nob;
output T3P[8]; T3P=noc;
output T4P[16]; T4P=nod;
output T5P[16]; T5P=noe;
output T6P[16]; T6P=nof;
output T7P[2]; T7P=sequence;
output T8P[2]; T8P=delaysequence;
output T9P; T9P=divres;
output T10P; T10P=eoc;
output T11P; T11P=eocb;
output T12P[4]; T12P=deg2p;
output T13P[4]; T13P=deg1p;
output T14P[4]; T14P=deg0p;
output T15P[2]; T15P=req;
output T16P; T16P=resseq;
output T17P; T17P=sequence2p;
output T18P; T18P=sequence3p;
output T19P[2]; T19P=ack;
output T20P; T20P=ireq;

{ ----- }
{  入力代入          }
{ ----- }
ireq=!REQ;

{ ----- }
{  出力代入          }
{ ----- }
BCD2P=deg2p;
BCD1P=deg1p;
BCD0P=deg0p;

ACK=!ack.0;

{ ----- }
{  行程 3 位置      }
{ ----- }
switch(sequence)
  case 3: sequence3p=1;
endswitch

{ ----- }
{  変換完了行程      }
{ ----- }
if (RESET)
  ack=0;
else
  switch(ack)
    case 0: if (sequence3p) ack=1; endif
    case 1: ack=2;
    case 2: if (sequence3p) ack=ack; else ack=0; endif
  endswitch
endif

{ ----- }

```

```

{   行程初期化   }
----- }
switch(req)
  case 1: resseq=1;
endswitch

{   変換前データ   }
----- }
addata.0:7=ADDATA;

{   行程 2 位置   }
----- }
switch(sequence)
  case 2: sequence2p=1;
endswitch

{   要求行程   }
----- }
if (RESET)
  req=0;
else
  switch(req)
    case 0:
      if (ireq) req=1; endif
    case 1: req=2;
    case 2:
      switch(sequence)
        case 3:
          if (ireq)
            req=req;
          else
            req=3;
          endif
        default: req=req;
      endswitch
    endswitch
  endif
endif

{   BCD1 桁   }
----- }
if (RESET)
  deg0p=0;
else
  if (sequence2p)
    if (eocb)
      deg0p=nod.0:3;
    else
      deg0p=deg0p;
    endif
  else
    deg0p=deg0p;
  endif
endif

{   BCD2 桁   }
----- }
if (RESET)
  deg1p=0;
else
  if (divres)

```

```

        deg1p=deg1p;
    else
        switch(sequence)
            case 1:
                if (eoc)
                    deg1p=noa.0:3;
                else
                    deg1p=deg1p;
                endif
            default: deg1p=deg1p;
        endswitch
    endif
endif
}
{-----}
{ BCD3 桁 }
{-----}
if (RESET)
    deg2p=0;
else
    if (divres)
        deg2p=deg2p;
    else
        switch(sequence)
            case 0:
                if (eoc)
                    deg2p=noa.0:3;
                else
                    deg2p=deg2p;
                endif
            default: deg2p=deg2p;
        endswitch
    endif
endif
}
{-----}
{ 計算 1 }
{-----}
if (!divres)
    noa=addata.0:7/nob-noc;
endif
}
{-----}
{ 計算 1 終了位置 }
{-----}
eoc=noa.8;
eoc.1=1;
}
{-----}
{ 計算値 1 代入 }
{-----}
noe.0:3=deg2p;
}
{-----}
{ 計算値 2 代入 }
{-----}
nof.0:3=deg1p;
}
{-----}
{ 計算 2 }
{-----}
if (sequence2p)
    nod=addata-noe*100-nof*10;
endif

```

```

{-----}
{ 計算 2 終了位置 }
{-----}
eocb=nod.16;
eocb.1=1;

{-----}
{ 主行程 }
{-----}
if (RESET|resseq)
sequence=0;
else
if (eoc)
if (divres)
sequence=sequence;
else
switch(sequence)
case 3: sequence=sequence;
default: sequence=sequence+1;
endswitch
endif
else
sequence=sequence;
endif
endif

{-----}
{ 遅延主行程 }
{-----}
delaysequence=sequence;

{-----}
{ 計算値 3 代入 }
{-----}
switch(sequence)
case 0: nob=100;
case 1: nob=10;
endswitch

{-----}
{ 計算値 4 代入 }
{-----}
switch(sequence)
case 0: noc=0;
case 1: noc=deg2p*10;
endswitch

{-----}
{ 割り算初期化 }
{-----}
if (RESET)
divres=1;
else
if (delaysequence!=sequence) divres=1; endif
endif

ende

```

```

{ ===== }
{ 機能実行譜 }
{ ===== }
entity sim
{ ----- }
{ 端子 }
{ ----- }
output RESET;
output REQ;
output ADDATA[8];
output BCDOP[4];
output BCD1P[4];
output BCD2P[4];
output ACK;
input simres;

{ ----- }
{ 検証計数 }
{ ----- }
bitr tc[8];

{ ----- }
{ 実効譜導入 }
{ ----- }
part main(RESET,REQ,ADDATA,BCDOP,BCD1P,BCD2P,ACK)

{ ----- }
{ 検証計数 }
{ ----- }
simres=0;
if (!simres) tc=tc+1; endif

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

{ ----- }
{ 変換前データ }
{ ----- }
if (tc<53)
ADDATA=123;
else
ADDATA=245;
endif

{ ----- }
{ 要求 }
{ ----- }
switch(tc)
case 60: REQ=0;
default: REQ=1;
endswitch

ende

endlogic

```