

```

//                                                    //
// Mercury-Unit version 4.0                               //
// Original circuit designed by nenetto, BEL, C.Yamauchi  //
//                                                    //
// ispLSI1016 TQFP44 "Lunar"                             //
// ispLSI modified and designed by C.Yamauchi (NTSC)      //
//                                                    //
// * PCM shift registers (16bit)                          //
// * OPNA interrupt vector                                //
// * OPNA sequence                                         //
// * OPNA reset operation                                 //
// * Output to X680x0 data bus                           //
//                                                    //

MODULE Lunar

    title 'Lunar'

    // Connect with 2096 //

    n384FS                pin 27 ;                " Y2
    SIN                   pin 30 ;                " IN2
    !S0, S1               pin 8,18 ;              " IN0,IN1
    AD1                   pin 40 ;                " IN3
    G1R                   pin 43 ;
    SOUT                  pin 13 istype 'reg' ;
    FMVR, FMVW            pin 44,12 ;

    // X680x0 bus //

    CLK10M                pin 5 ;                " Y0
    CLK10M_               pin 29 ;                " Y1
    D0, D1, D2, D3        pin 19,20,21,22 ;
    D4, D5, D6, D7        pin 23,24,25,26 ;
    D8, D9, D10, D11      pin 31,32,33,34 ;
    D12, D13, D14, D15    pin 35,36,37,38 ;
    R_W_                  pin 9 ;
    !IACKx                pin 10 ;
    !IRQx                 pin 11 istype 'reg' ;
    !LDS                  pin 2 ;

    !EXRESET              pin 14 ;

    // OPNA //

    !IRQM                 pin 16 ;

    LROM                  pin 15 ;
    LROS                  pin 4 ;
    !ICM                  pin 3 istype 'reg' ;
    !ICS                  pin 42 ;

    !RD, !WD              pin 1,41 ;

    // node //

    // Shift registers //
    SR1, SR2, SR3, SR4    node istype 'reg' ;
    SR5, SR6, SR7, SR8    node istype 'reg' ;
    SR9, SR10, SR11, SR12 node istype 'reg' ;
    SR13, SR14, SR15      node istype 'reg' ;

    // OPNA inerrupt vector //

    FMV0, FMV1, FMV2, FMV3 node istype 'reg' ;
    FMV4, FMV5, FMV6, FMV7 node istype 'reg' ;

    DIRQM, PIRQM, DIACKx  node istype 'reg' ;

```

```

IRQST, IRQED          node  istype 'com' ;

DP1, DP2, DP3          node  istype 'reg' ;
DPLSFM                node  istype 'com' ;

PICS                   node  istype 'com,keep' ;

DOE                    node  istype 'com' ;

PEXTRST, EXTRST       node  istype 'reg' ;

```

```

plsi property 'SNP LROS path_1' ;
plsi property 'ENP PICS path_1' ;
plsi property 'PRESERVE PICS' ;

plsi property 'SNP LROS path_2' ;
plsi property 'ENP ICS path_2' ;

plsi property 'Y1_AS_RESET OFF' ;

plsi property 'PULLUP OFF' ;

plsi property 'PULLUP IACKx' ;
plsi property 'PULLUP IRQM' ;
plsi property 'PULLUP LROM' ;
plsi property 'PULLUP LROS' ;

```

Equations

```

//      PCM      //

//      Shift registers      //

SR1      :=  SIN & S0 & !S1
          #   D0 & S0 & S1
          #   SR1 & !S0 ;
SR1.clk  =  n384FS ;

SR2      :=  SR1 & S0 & !S1
          #   D1 & S0 & S1
          #   SR2 & !S0 ;
SR2.clk  =  n384FS ;

SR3      :=  SR2 & S0 & !S1
          #   D2 & S0 & S1
          #   SR3 & !S0 ;
SR3.clk  =  n384FS ;

SR4      :=  SR3 & S0 & !S1
          #   D3 & S0 & S1
          #   SR4 & !S0 ;
SR4.clk  =  n384FS ;

SR5      :=  SR4 & S0 & !S1
          #   D4 & S0 & S1
          #   SR5 & !S0 ;
SR5.clk  =  n384FS ;

SR6      :=  SR5 & S0 & !S1
          #   D5 & S0 & S1
          #   SR6 & !S0 ;
SR6.clk  =  n384FS ;

SR7      :=  SR6 & S0 & !S1
          #   D6 & S0 & S1
          #   SR7 & !S0 ;
SR7.clk  =  n384FS ;

```

```

SR8      :=  SR7 & S0 & !S1
           #   D7 & S0 & S1
           #   SR8 & !S0 ;
SR8.clk  =  n384FS ;

SR9      :=  SR8 & S0 & !S1
           #   D8 & S0 & S1
           #   SR9 & !S0 ;
SR9.clk  =  n384FS ;

SR10     :=  SR9 & S0 & !S1
           #   D9 & S0 & S1
           #   SR10 & !S0 ;
SR10.clk =  n384FS ;

SR11     :=  SR10 & S0 & !S1
           #   D10 & S0 & S1
           #   SR11 & !S0 ;
SR11.clk =  n384FS ;

SR12     :=  SR11 & S0 & !S1
           #   D11 & S0 & S1
           #   SR12 & !S0 ;
SR12.clk =  n384FS ;

SR13     :=  SR12 & S0 & !S1
           #   D12 & S0 & S1
           #   SR13 & !S0 ;
SR13.clk =  n384FS ;

SR14     :=  SR13 & S0 & !S1
           #   D13 & S0 & S1
           #   SR14 & !S0 ;
SR14.clk =  n384FS ;

SR15     :=  SR14 & S0 & !S1
           #   D14 & S0 & S1
           #   SR15 & !S0 ;
SR15.clk =  n384FS ;

SOUT     :=  SR15 & S0 & !S1
           #   D15 & S0 & S1
           #   SOUT & !S0 ;
SOUT.clk =  n384FS ;

```

// Output to X680x0 data bus //

```

DOE      =  G1R # FMVR ;

D0       =  G1R & SR1 #
           FMVR & FMV0 ;
D0.oe    =  DOE ;

D1       =  G1R & SR2 #
           FMVR & FMV1 ;
D1.oe    =  DOE ;

D2       =  G1R & SR3 #
           FMVR & FMV2 ;
D2.oe    =  DOE ;

D3       =  G1R & SR4 #
           FMVR & FMV3 ;
D3.oe    =  DOE ;

D4       =  G1R & SR5 #
           FMVR & FMV4 ;
D4.oe    =  DOE ;

```

```

D5          = G1R & SR6 #
              FMVR & FMV5 ;
D5.oe       = DOE ;

D6          = G1R & SR7 #
              FMVR & FMV6 ;
D6.oe       = DOE ;

D7          = G1R & SR8 #
              FMVR & FMV7 ;
D7.oe       = DOE ;

D8          = !G1R # SR9 ;
D8.oe       = DOE ;

D9          = !G1R # SR10 ;
D9.oe       = DOE ;

D10         = !G1R # SR11 ;
D10.oe      = DOE ;

D11         = !G1R # SR12 ;
D11.oe      = DOE ;

D12         = !G1R # SR13 ;
D12.oe      = DOE ;

D13         = !G1R # SR14 ;
D13.oe      = DOE ;

D14         = !G1R # SR15 ;
D14.oe      = DOE ;

D15         = !G1R # SOUT ;
D15.oe      = DOE ;

//      OPNA interrupt vector (Registers)      //

FMV0        := ( FMVW & D0 # !FMVW & FMV0 ) # EXRESET ;
FMV0.clk    = CLK10M_ ;

FMV1        := ( FMVW & D1 # !FMVW & FMV1 ) # EXRESET ;
FMV1.clk    = CLK10M_ ;

FMV2        := ( FMVW & D2 # !FMVW & FMV2 ) # EXRESET ;
FMV2.clk    = CLK10M_ ;

FMV3        := ( FMVW & D3 # !FMVW & FMV3 ) # EXRESET ;
FMV3.clk    = CLK10M_ ;

FMV4        := ( FMVW & D4 # !FMVW & FMV4 ) # EXRESET ;
FMV4.clk    = CLK10M_ ;

FMV5        := ( FMVW & D5 # !FMVW & FMV5 ) # EXRESET ;
FMV5.clk    = CLK10M_ ;

FMV6        := ( FMVW & D6 # !FMVW & FMV6 ) # EXRESET ;
FMV6.clk    = CLK10M_ ;

FMV7        := ( FMVW & D7 # !FMVW & FMV7 ) # EXRESET ;
FMV7.clk    = CLK10M_ ;

//      OPNA sequence      //

DP1         := AD1 ;
DP1.clk     = CLK10M ;

DP2         := DP1 ;

```

```

DP2.clk      = CLK10M ;

DP3          := DP2 ;
DP3.clk      = CLK10M ;

DPLSFM       = DP1 & !DP3 ;

RD           = R_W_ ;

WD           = !R_W_ & LDS & DPLSFM ;

// Interrupt //

DIRQM        := IRQM ;
DIRQM.clk    = CLK10M ;

PIRQM        := DIRQM ;
PIRQM.clk    = CLK10M ;

IRQST        = DIRQM & !PIRQM ;

DIACKx       := IACKx ;
DIACKx.clk   = CLK10M ;

IRQED        = !DIACKx & IACKx ;

IRQx         := IRQST # IRQx & !IRQED & !EXRESET ;
IRQx.clk     = CLK10M ;

// OPNA reset operation //

PEXTRST      := EXRESET ;
PEXTRST.clk  = !LROM ;

EXTRST       := EXRESET # PEXTRST ;
EXTRST.clk   = !LROM ;

ICM          := ( EXTRST # EXRESET ) ;
ICM.clk      = !LROM ;

PICS         = ( EXRESET # ICM ) & LROS #
              PICS & !LROS ;

ICS          = PICS & !LROS #
              ICS & LROS ;

END

```