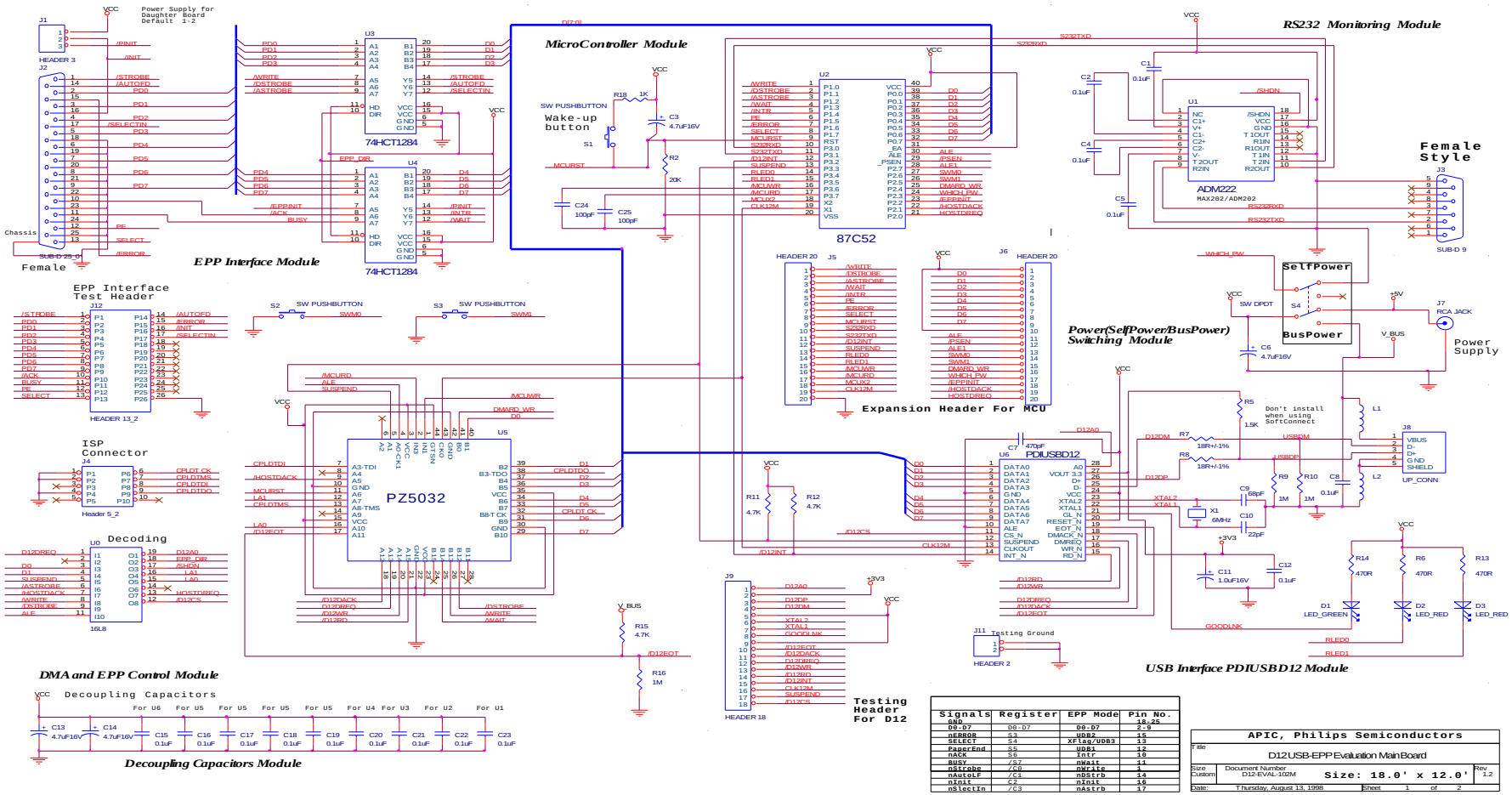

Figure 4-3: Schematic of the main board.

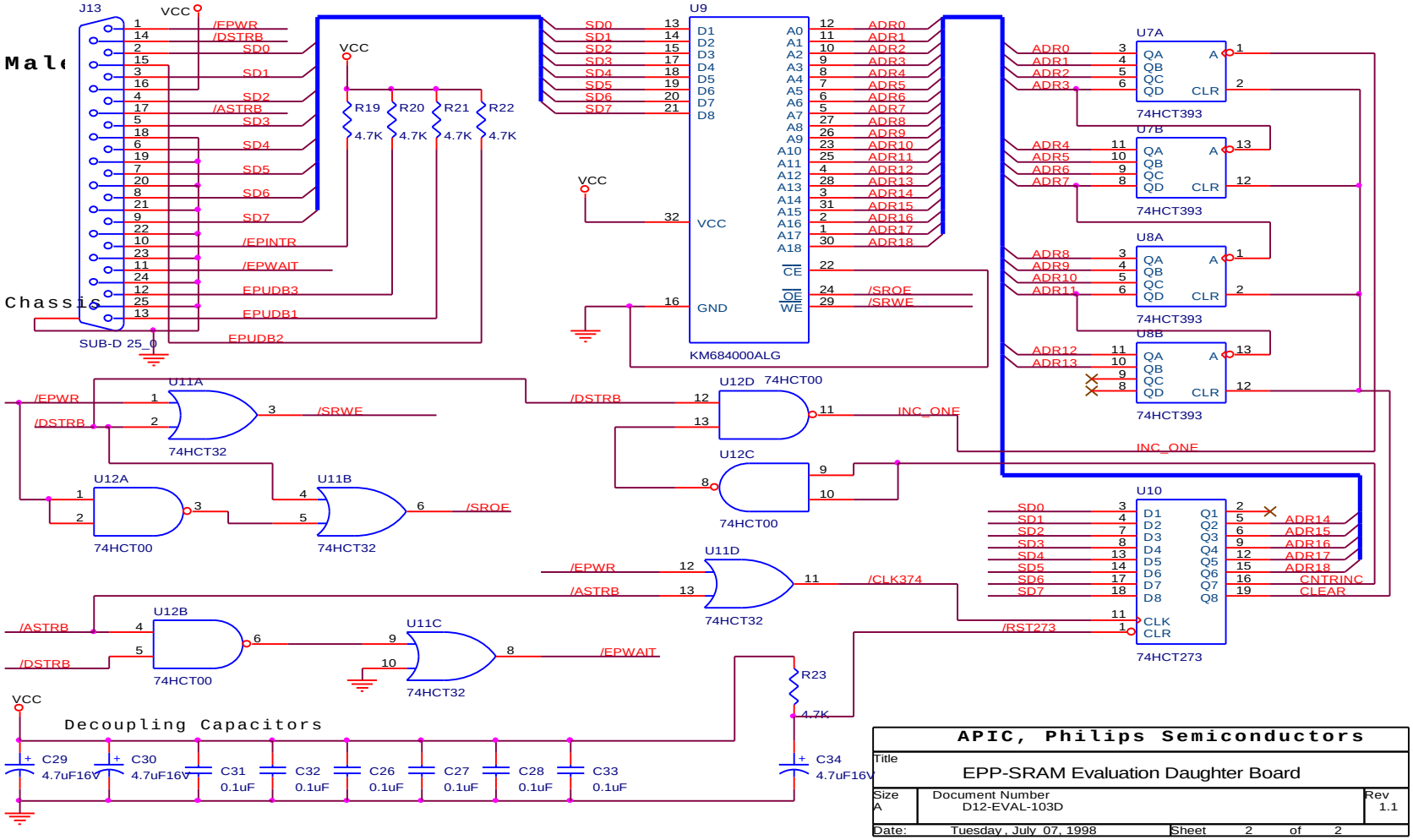

Figure 4-4: Schematic of the daughter board

4.4 Appendix D: BOM (Bill of Materials)

4.4.1 Main Board's BOM

S/N	Item P/N & Description	Reference	Qty	Package	Approved Vendors	Remark
M.1	PDIUSBD12, Philips USB Interface device with parallel	U6	1	DIP28(600 mil) & SO28	Philips	
M.2	PZ5032CSA44, Philips 32 Micro-Cell CPLD	U5	1	PLCC44	Philips	
M.3	P87C52UBPN, Philips 8051 compatible Micro-Controller	U2	1	DIP40, 600mil	Philips	
M.4	74HCT1284D, Philips IEEE1284 buffer	U3,U4	2	SOL20, 20-pin plastic	Philips	
M.5	ADM222/ADM202 , CMOS RS-232 Driver/Receiver	U1	1	SOIC16(ADM202)/SOIC18(A	Analog Devices	
M.6	16L8-15,PAL device	U0	1	DIP20,300 mil	Open Source	
M.7	6.0 Mhz Crystal, TTL or CMOS Output with 0.01% Frequency Tolerance, 40-60% Duty Cycle	X1	1	Thru' hole ,Low Profile, Lead's Pitch =0.3"	Open Source	
M.8	Push –Button, Normally open	S1,S2,S3	3	Thru' hole	Open Source	
M.9	Double Pole and Double Thread Switch, DPDT	S4	1	Thru' Hole	Open Source	
M.10	Green Color LED	D1	1	Thru' Hole, Lead's Pitch =0.1"	Open Source	
M.11	Red Color LED	D2,D3	2	Thru' Hole, Lead's Pitch =0.1"	Open Source	
M.12	DB 9-pin/Female D-Type/Right angle PCB Mount	J3	1	Thru' Hole	Open Source	
M.13	DB 25pin/ Female D-Type/Right Angle PCB Mount	J2	1	Thru' Hole	Open Source	
M.14	RCA Jack for Self Power supply(+5V)	J7	1	Thru' Hole	Open Source	
M.15	USB B-Type Connector(Upstream)	J8	1	Thru' Hole	Open Source	
M.16	Single Row, 2-Pin Straight Header, Solder Type	J11	1	SIL2, Lead's Pitch =0.1"	Open Source	
M.17	Single Row, 3-pin Straight Header, Solder Type	J1	1	SIL3, Lead's Pitch =0.1"	Open Source	
M.18	Single Row,18-pin Straight Header ,Solder Type	J9	1	SIL18, Lead's Pitch =0.1"	Open Source	
M.19	Single Row, 20-Pin Straight Header, Solder Type	J5,J6	2	SIL20, Lead's Pitch =0.1"	Open Source	
M.20	Double Row, 10-Pin Straight Header, Solder Type	J4	1	DIL10, Lead's Pitch =0.1"	Open Source	
M.21	Double Row, 26-Pin Straight Header, Solder Type	J12	1	DIL26, Lead's Pitch =0.1"	Open Source	
M.22	Ferrite Bead, Multi-layer Chip Bead(10nH)	L1,L2	2	1206	Open Source	
M.23	18R resistor, 1/10W,+/-1%	R7,R8	2	0805	Open Source	
M.24	470R resistor, 1/10W,+/-5%	R6,R13,R14	3	0805	Open Source	
M.25	1.0K resistor, 1/10W,+/-5%	R18	1	0805	Open Source	
M.26	1.5K resistor, 1/10W,+/-5%	R5	1	0805	Open Source	
M.27	4.7K resistor, 1/10W,+/-5%	R11,R12,R15	3	0805	Open Source	
M.28	20K resistor, 1/10W,+/-5%	R2	1	0805	Open Source	
M.29	1M resistor, 1/10W,+/-5%	R9,R10,R16	3	0805	Open Source	
M.30	22pF Capacitor, 50V, Multilayer,+/-20%	C10	1	0805	Open Source	
M.31	68pF Capacitor, 50V,Multilayer, +/-20%	C9	1	0805	Open Source	

M.32	100pF Capacitor, 50V, Multi-layer, +/-20%	C24,C25	2	0805	Open Source	
M.33	470pF Capacitor, 50V, Multi-layer, +/-20%	C7	1	0805	Open Source	
M.34	0.1uF Capacitor, 50V,Multi-layer ,+/-20%	C1,C2,C4,C5,C8,C12,C15,C16, C17,C18, C19,C20,C21,C22,C23,C24,	16	0805	Open Source	
M.35	1.0uF Capacitor, 16V, Electrolytic/multi-layer Chip Bead,	C11	1	Case "C"	Open Source	
M.36	4.7uF Capacitor, 16V, +/-20%,Electrolytic/Multilayer	C3,C6,C13,C14,	4	Case "C"	Open Source	
M.37	2-Layer PCB, USB-EPP Main Board, Size 5.3" *4.25"	2 Layers	1	Rectangle	ESA	

Table 4-10: BOM for main board.

4.4.2 Daughter Board's BOM

S/N	Item P/N & Description	Reference	Qty	Package	Approved Vendors	Remark
D.1	74HCT393, CMOS Dual 4-Bit Ripple Counter	U7,U8	2	SOIC14	Open Source	
D.2	KM6840000ALG,512K*8Bit SRAM	U9	1	32-Pin SOP	SamSung	
D.3	74HCT273 , Octal D-Type Flip-Flop With Reset	U10	1	SOIC20	Open Source	
D.4	74HCT32,Quad 2-Input OR Gate	U11	1	SOIC14	Open Source	
D.5	74HCT00,Quad 2-Input NAND Gate	U12	1	SOIC14	Open Source	
D.6	DB 25pin Male D-Type/Right angle Connector	J13	1		Open Source	
D.7	4.7K resistor, 1/10W,+/-5%	R19,R20,R21,R22,R23	1	0805	Open Source	
D.8	0.1uF Capacitor, 50V, Multiu-layer, +/-20%	C26,C27,C28,C31,C32,C33	1	0805	Open Source	
D.9	4.7uF Capacitor, 16V, Electrolytic or Multi-layer Chip	C29,C30,C34	3	Case "C"	Open Source	
D.10	2-Layer PCB, USB-EPP Daughter Board, Size 1.75" *	2 Layers	1	Rectangle	ESA	

Table 4-11: BOM for daughter board.

4.5 Appendix E: CPLD VHDL Simulation Waveforms

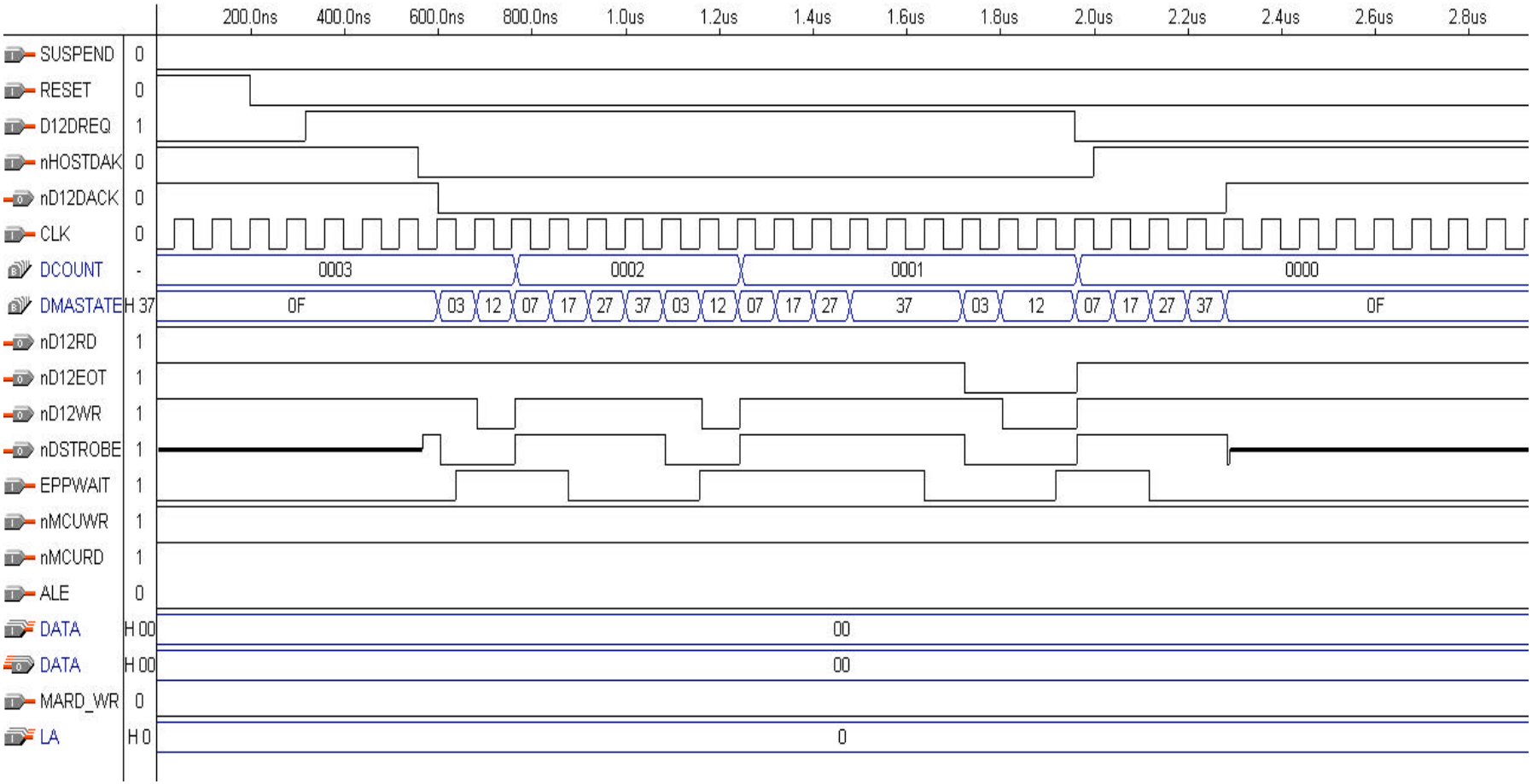

Figure 4-5: DMA Write-Timing (Reading from EPP and Writing to D12).

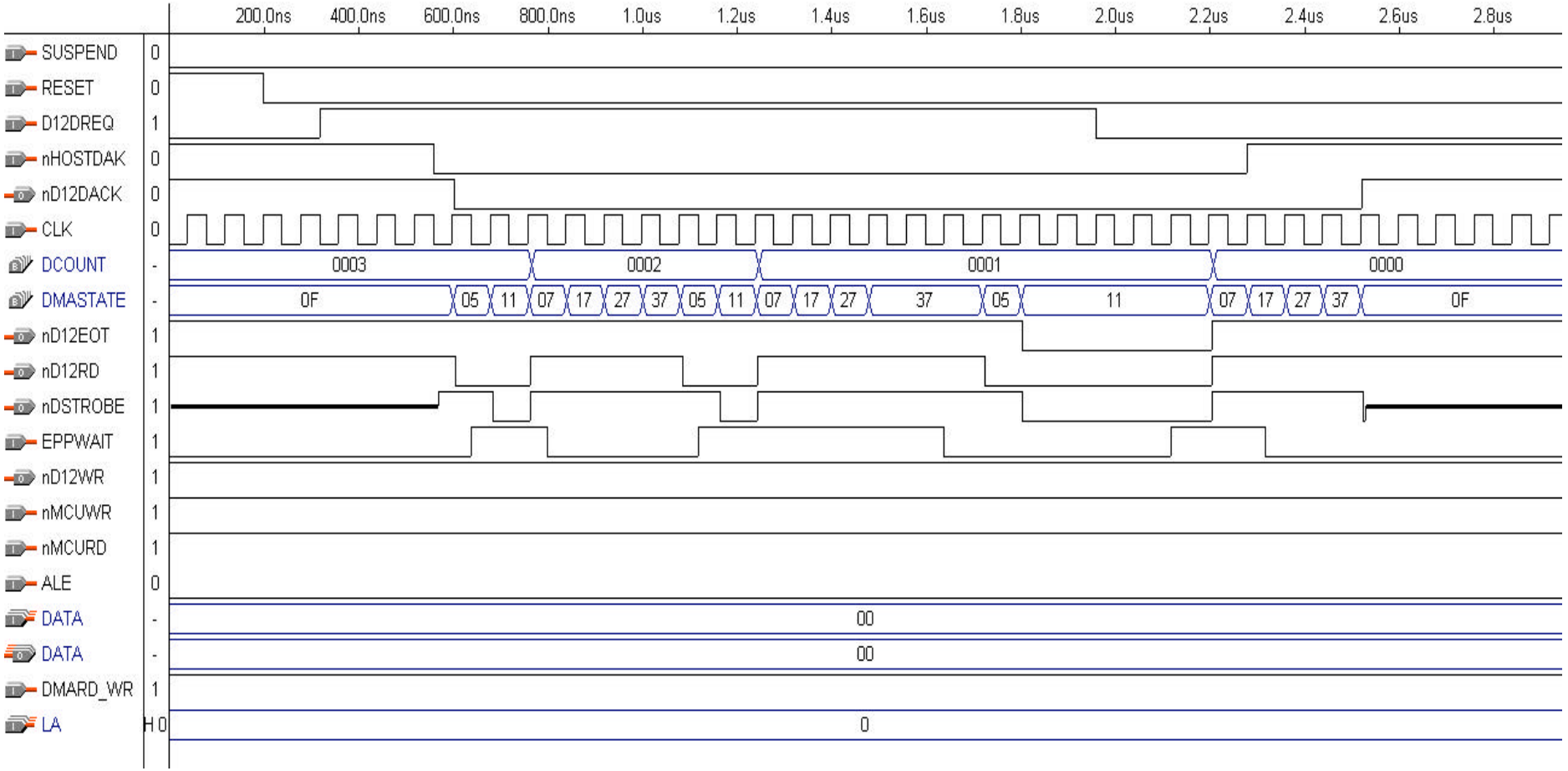

Figure 4-6: DMA Read-Timing (Reading from D12 and Writing to EPP)

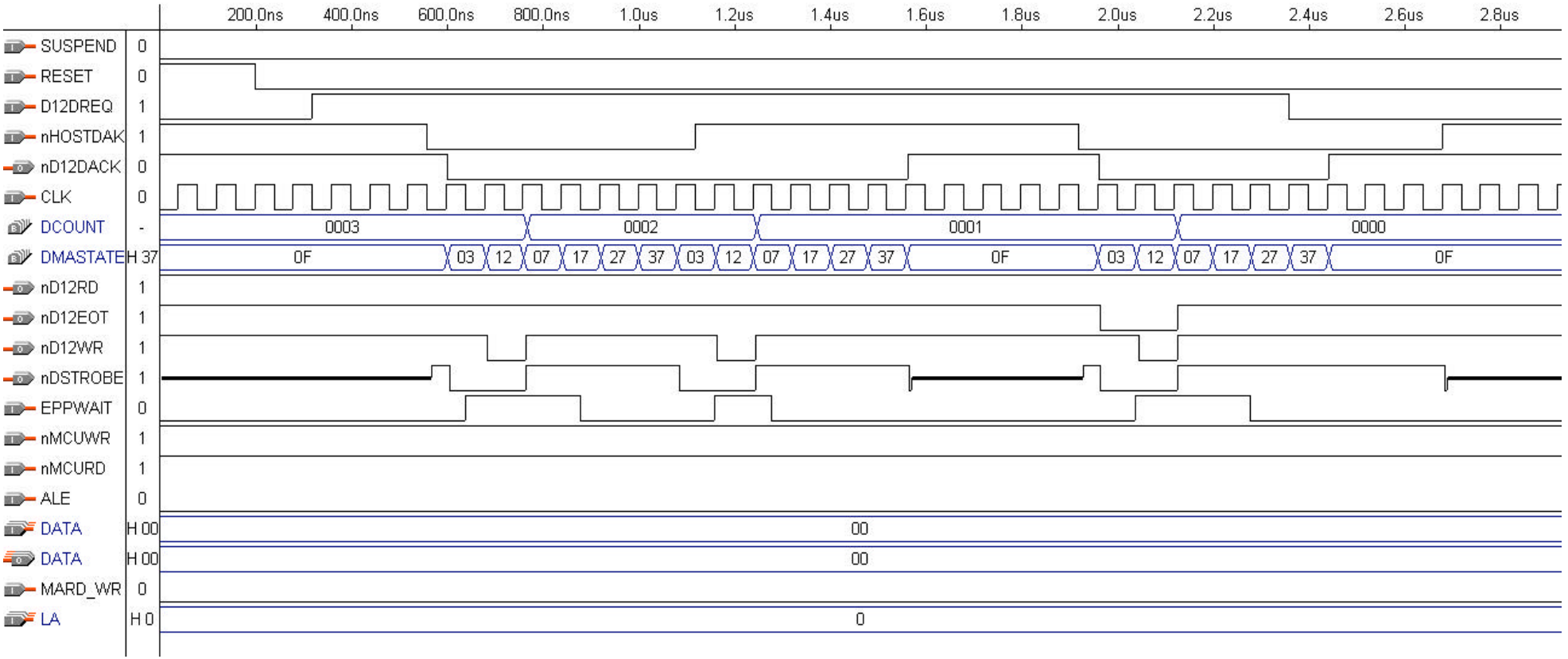

Figure 4-7: DMA Read-Timing (Reading from D12 and Writing to EPP) with nHOSTDAK inactive during DMA transfer.

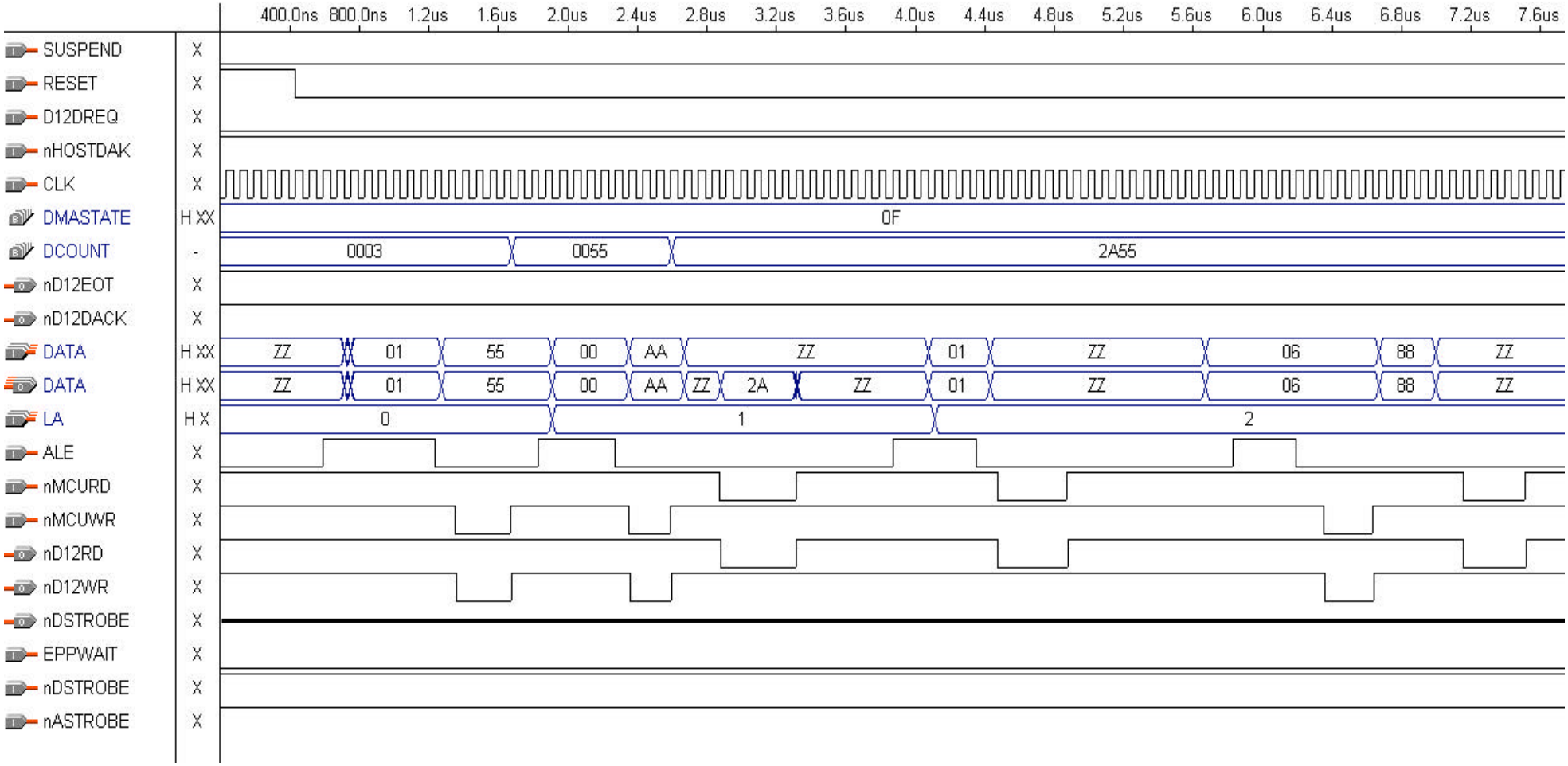

Figure 4-8: MCU accessing CPLD and D12 timing.

4.6 Appendix F: CPLD VHDL Source Codes

```
--*************************************************************************
--
-- ProjectName :USB-EPP Evaluation Kit
--  Chip(CPLD)Name: DMA-EPP Controller
--  VHDL Code:    EPPDMA.VHD (simplified version for State Machine)
--  Target CPLD:PZ5032cs10A44
-- Designer:    Steven Cheng
-- Copy Right:  APIC, Philips (Semiconductors) Singapore
--   Last updated:Aug.17 , 1998
--  Version;   1.02
--*************************************************************************
--
library ieee; -- IEEE general Library
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
--library ASYL; --Philips CPLD library
--use ASYL.SL_ARITH.all;
--use ASYL.ARITH.all;
-------------------------------------------------------------------------------
entity EPPDMA is
    port(
   CLK :      in      std_logic; common Clock 12Mhz
   RESET:     in      std_logic; -- high active
   DATA:   inout   std_logic_vector(7 downto 0); -- data bus
   LA:        in      std_logic_vector(1 downto 0);--latched address bus
   DMARD_WR: in      std_logic; -- DMA transfer direction, 1=read from
D12


   ---D12 Signals -----------
   D12DREQ:   in    std_logic; -- D12 DMA request
   nD12WR:    buffer  std_logic; -- D12 Write
   nD12RD:    buffer  std_logic; -- D12 Read
   nD12DACK: buffer  std_logic; -- D12 DMA ACK
   nD12EOT:   out   std_logic; -- D12 End of Transfer
   --SUSPEND:in    std_logic; -- Suspend mode

   ----MCU Signals -----------
   nMCUWR:    in    std_logic; -- MCU write
   nMCURD:    in    std_logic; -- MCU read
   ALE:    in    std_logic; -- Address latch Enable
   nHOSTDAK: in    std_logic; -- Host DMA ACK

   ----EPP Signals
   EPPWAIT:in    std_logic; -- EPP Wait
   nDSTROBE: out  std_logic -- EPP Data Strobe
   --nEPPWR:    out     std_logic; -- EPP Write,
   );

end EPPDMA;
-------------------------------------------------------------------------------

architecture behave of EPPDMA is
-- LA="00">- CPLD internal Register Low;
-- LA="01">- CPLD Internal Register High;
-- LA="10">- D12 Data Register;
-- LA="11">- D12 Command Register;
--*************************************
--------State Machine Definition
   --Trueth Table
   -- State   DELAY1  DELAY0 nD12DACK nDSTROB    nD12RD    nD12WR
   -- IDLE      0     0     1     1       1       1
   -- DMAREAD1  0     0     0     1       0       1
   -- DMAREAD2  0     1     0     0       0       1


   --DMAWRIT1  0     0     0     0       1       1
   -- MMAWRIT2  0     1     0     0       1       0

   -- DMAEND    0     0     0     1       1       1
   -- DMAEND1   0     1     0     1       1       1
   -- DMAEND2   1     0     0     1       1       1
   -- DMAEND3   1     1     0     1       1       1


   ----DCOUNT Parameters
   Constant BITW: integer= 13; -- Width of CPLD internal Counter-1
   Constant BITWZERO:std_logic_vector(15 downto (BITW+1)):="00"; --
   Constant BITWZERO1:std_logic_vector(BITWdownto 0):="0000000000011"; -
-
   Constant BITWZERO2:std_logic_vector(BITWdownto 1):="000000000000"; --

   signal DOUT:    std_logic_vector (7downto 0); --Data output
   --signal DELAY:    integer range  0 to 3; --delay cycle for D12 READ
High

    signal DIN:    std_logic_vector (7downto 0); --Data Input
    signal DCOUNT:    std_logic_vector (BITWdownto 0); --CPLD Internal
Counter

   signal DMA_ON: std_logic; -- DMA is on Cycle --Aug 8
           signal RWCLK:  std_logic; -- Clock for internal Counter
```

```
   signal DCZERO: std_logic; -- Counter return to Zero
   signalnDSTROB:    std_logic; -- EPP Data Strobe


   signal DMASTATEstd_logic_vector(5downto 0); -- State Machine

   Constant IDLE: std_logic_vector(5 downto 0):="001111"; --IDLE state
   Constant DMAREAD1:std_logic_vector(5 downto 0):="000101"; -- DMA read
state 1
   Constant DMAREAD2:std_logic_vector(5 downto 0):="010001"; -- DMA read
State 2

   Constant DMAWRIT1:std_logic_vector(5 downto 0):="000011"; -- DMA write
State 1
   Constant DMAWRIT2:std_logic_vector(5 downto 0):="010010"; -- DMA write
State 2

   Constant DMAENDstd_logic_vector(5 downto 0):="000111"; -- Current DMA
R/W End
   Constant DMAEND1std_logic_vector(5 downto 0):="010111"; -- Current DMA
R/W End
   Constant DMAEND2std_logic_vector(5 downto 0):="100111"; -- Current DMA
R/W End
   Constant DMAEND3std_logic_vector(5 downto 0):="110111"; -- Current DMA
R/W End


-------------------------------------------------------------------------
----


begin

---
*************************************************************************
*
STATE: process (RESET,CLK)
begin
   if RESET='1' then
        DMASTATE<=IDLE;

   elsifCLK'event and CLK='1' then
   ---------------------------------------------
      --- State Transition -----
        Case  DMASTATE  is

        ------IDLE state --------
       when IDLE =>
```

```
 if nHOSTDAK='0' and EPPWAIT='0'  then
      --if D12DREQ='1' and (DCZERO='0' or DCOUNT(0)='1') then
      if D12DREQ='1'  then
         if DMARD_WR='1' then
            DMASTATE<=DMAREAD1; --- DMA Read Transfer

         else
            DMASTATE<=DMAWRIT1; -- DMA write Transfer

         end if;
      end if;
end if;


---DMA read Cycle--------------------
when DMAREAD1 =>
   --if  EPPWAIT='1' then
      DMASTATE<= DMAREAD2; -- Read
   --end if;


when DMAREAD2 =>
   if  EPPWAIT='1' then
      DMASTATE<= DMAEND; -- Current DMA transfer finished
   end if;
-------------------------------------


---DMA read Cycle----------------------
when DMAWRIT1 =>
   DMASTATE<= DMAWRIT2; -- write

when DMAWRIT2 =>
   if EPPWAIT='1' then
      DMASTATE<= DMAEND; -- Current DMA transfer finished

   end if;
-------------------------------------


when DMAEND =>
      DMASTATE<= DMAEND1; -- Current DMA transfer finished

when DMAEND1 =>
      DMASTATE<= DMAEND2; -- Current DMA transfer finished
```

```
        when DMAEND2 =>
             DMASTATE<= DMAEND3; -- Current DMA Transfer finished

        when DMAEND3 =>
           if nHOSTDAK='0' and EPPWAIT='0' and D12DREQ ='1'
              and (DCZERO ='0' or DCOUNT(0)='1') then

              if DMARD_WR='1' then --
                 DMASTATE<= DMAREAD1; -- Continue the Burst read mode

              else --
                 DMASTATE<= DMAWRIT1; -- continue the Burst Write Mode
              end if;

           elsif EPPWAIT='0' then


              DMASTATE<= IDLE; -- Idle state

           end if;


        when OTHERS =>
       end case;

   end if;

end process STATE;
---
*************************************************************************
*



---Internal Counter
operation*************************************************
counter: process(RWCLK,RESET)
begin
   if RESET='1' then
      DCOUNT(BITWdownto 0) <= BITWZERO1; --Init value


   elsifRWCLK'event and RWCLK='1' then
      if DMA_ON ='1' then  -- DMA on Cycle
          DCOUNT<=DCOUNT-'1'; --after transfer byte one counter -1

      else -- DMA off Cycle
```

```
          case  LA is  -- Address input
             when "00" => -- Address 00
                DCOUNT(7downto 0) <=DIN(7downto 0); -- Counter Low Byte

             when "01" => -- Address 01
                DCOUNT(BITWdownto 8) <=DIN((BITWdown0)nto 0);--Counter High
Byte

             when others =>

          end case;
       end if;
    end if;
end process counter;
---
*************************************************************************
*



---
*************************************************************************
*
---
*************************************************************************
*
--Databus  for read/write from/to internal counter
DOUT<=  DCOUNT(7downto 0) when LA="00" else
BITWZERO&  DCOUNT(BITWdownto 8) ;

--- Decoding Logic ---------------------------------------
----States and D12 control signals and EPP signals -----
   nD12DACK<=DMASTATE(3);-- D12ACK DMA

   -- Internal Dstrobe signal
   nDSTROB<=DMASTATE(2);-- Internal /DSTROBE

   -- External Dstrobe signal
   nDSTROBE<=DMASTATE(2) when DMA_ON='1'  else 'Z';

   --- D12 read/Write control
   nD12RD  <=DMASTATE(1) when DMA_ON='1' else nRD;--
   nD12WR  <=DMASTATE(0) when DMA_ON='1' else nWR;--

   ----EPP Write signals -----
   --nEPPWR<= not DMARD_WR when DMA_ON='1' else 'Z' ;
```

```
   ----DMA End of Transfer Signal ----
   nD12EOT <='0' when nDSTROB='0' and DCZERO='1' and DCOUNT(0)='1' else
'1';

   ---- Counter decreasement or preset(load) Clock-----
   RWCLK<='0' when (nDSTROB='0' and nD12DACK='0') or
        (nMCUWR='0' and nD12DACK='1')  else '1';  --

   ---Control return to 0/1 control ------
   DCZERO  <='1' when DCOUNT(BITW downto 1) = BITWZERO2 else '0';

   ---Data  Input/output Bus control ------
   DIN     <= DATA;
   DATA <= DOUT when nMCURD='0' and ALE='0' and (nD12DACK='0') else "ZZZZZZZZ";

   ---DMA_On means that DMA Cycle is carrying on when nHOSTDAK is high
   DMA_ON <='0' when (nHOSTDAK='1' and nD12DACK='1') else '1';
-------------------------------------------------------------------------------
end behave;
```

4.7 Appendix G -

Main Board's PAL (16L8) Equations

As the 32 micro-cells CPLD is not enough to fit the following PAL equations, one additional PAL device (Lattice or AMD PAL18L8) is used to implement this simple coding. The related signals (inputs/outputs) are contained in Figure3-4. The equations are as follows:

```
Name     USBEPP;
Partno   CA0016;
Date     06/07/98;
Rev      01;
Designer  Steven Cheng;
Company   Philips Semiconductors ;
Assembly  None;
Location  None;
Device    g16V8;
```

```
/*****************************************************************/
/*                                                               */
/* USB-EPP Evaluation Board decoding                             */
/*                                                               */
/*****************************************************************/
/* Allowable Target Device Types :  PAL16L8                      */
/*****************************************************************/
/** Inputs **/
Pin 1 = D12DREQ; /* D12 DMA request */
/*Pin 2 = DMA_ON; */
Pin 3 = D0;      /* Data Bus D0 */
Pin 4 = D1;        /* Data Bus D1*/
Pin 5 = SUSPEND; /* D12 Suspend Mode */
Pin 6 = ASTROBE; /* EPP ASTROBE */
Pin 7 = HOSTDAK; /* Host DMA ACK */
Pin 8 = EPPWR; /* EPP Write */
Pin 9 = DSTROBE; /* EPP DSTROBE */
Pin 11 = ALE;  /* ADdress Latch */

/** Outputs **/
pin 12 = D12CS; /* D12 Chip select */
Pin 13 = HOSTDREQ;    /* Host DMA Request      */
/*Pin 14 = DSTROBE    EPP DSTROBE, MCU Control */
Pin 15 = LA0;         /* Address A0                  */
Pin 16 = LA1;         /* Address A1    */
pin 17 = SHDN; /* ADM222 Shut down */
Pin 18 = EPPDIR;  /* /*  EPP 1284 buffer direction control */
Pin 19 = D12A0;   /* D12 Address A0*/

/*pin 19 = */

/* Adder-slice circuit - add 2, 1-bit, numbers with carry */
/* Perform 4, 1-bit, additions and keep the final carry */

LA0 = (D0 & ALE) # (LA0 & !ALE) ;
LA1 = (D1 & ALE) # (LA1 & !ALE) ;
HOSTDREQ = D12DREQ;
D12A0= LA0 & LA1 & !ALE ;
!D12CS  = LA1 & !ALE & HOSTDAK;
!SHDN= SUSPEND;
EPPDIR  = (EPPWR & !DSTROBE) #
(HOSTDAK & EPPWR & !ASTROBE) ;
```