Logical Interface Futurebus+ Engine (LIFE) Design for a MC68040 Based Board

National Semiconductor
Application Note 836
Webster (Rusty) B. Meier Jr.
June 1992

I. INTRODUCTION

This application note describes a Futurebus+ board design based upon the MC68040 CPU and the National Semiconductor LIFE Protocol Controller. This application note contains:

— A schematic diagram of the devices simulated;
— GAL® programmable logic equations for the GAL devices used, and;
— Simulation timing of the design.

Currently this design has been simulated using the Verilog modeling language. All Verilog models for the components used in this simulation were written by the author. It is assumed that the reader is familiar with the MC68040 microprocessor, GAL/PAL® design, and the LIFE Protocol and Data Path Unit. But, I will cover the LIFE Local Bus Protocol in some detail within this application note.

II. DESIGN DESCRIPTION

As stated above component models were written in Verilog and used to simulate this design. The component models that have been written and used in this simulation are as follows:

- MC68040 microprocessor model, this component models the MC68040 bus timing including:
  - — Read accesses, single and burst,
  - — Write accesses, single and burst,
  - — Transfer Burst Inhibit (TBI__) capability,
  - — Local bus arbitration including Bus Request (BR__), Bus Grant (BG__), and Bus Busy (BB__),
  - — The insertion of wait states to the CPU (TA__),
  - — The ability to perform a bus relinquish and retry of the present bus transaction (TA__, TEA__);
- LIFE Protocol Controller model. This model allows the user to simulate the LIFE local bus timing for:
  - — LIFE (as local bus master) read and write accesses (single and burst),
  - — Host bus (as local bus master) read and write accesses (single and burst) to Futurebus+,
  - — Host bus (as local bus master) read and write accesses to the LIFE internal registers;
- 74F543 Octal Registered transceiver;
- 74xxx863 (National Semiconductor 74ALS29863) 9-bit bus transceiver;
- GAL16V8 programmable logic device used as the local bus arbiter and as a command interpreter between the LIFE and the local bus (MC68040 type protocol);
- GAL22V10 programmable logic device as the interface between the LIFE and the local bus (MC68040 type protocol);
- mem__blk, this block simulates a memory device that can be read and written to, and supplies a data acknowledge back to the local bus via the DTACK__ signal.

This design is a simulation of the local bus, of a Futurebus+ module, during:

- Read and write (single and burst) transactions across Futurebus+ with this module acting as a slave module;
- Read and write (single and burst) transactions across Futurebus+ with this module acting as the master module;
- Read and write transactions to the LIFE internal registers.

Several of the signals in this design should have pull-up resistors, (i.e., LAD<31:0>, LBP<3:0>, LCP), these were provided in the simulation input file.

Both the LIFE and the MC68040 Verilog models were written to provide typical local bus timing. The user should run through any critical timing paths to guarantee that the design will work at his particular clock speed, device type, and performance requirements.

Since these models (LIFE, MC68040) were written to provide local bus timing, the models work similar to a DMA controller. The user must program in the address, whether read, write or locked access, the number of accesses to perform, etc., and then set a ''go'' bit in the model. This will cause the particular model to begin the transfer desired. The model will automatically try to perform burst accesses if possible. Note that the 68040 is limited to a maximum burst access of 4 transfers.

It will be noticed that this design allows a maximum of one data transfer (32 bits plus parity on the local bus) for every two bus clocks. Both the MC68040 and the LIFE allow data to be transferred every bus clock cycle. This transfer rate could be hard to sustain during MC68040 write cycles at clock speeds above 20 MHz because of the data valid time for data out of the MC68040, 74xxx863 delay, and the required data setup time to the LIFE.

Equation #1, Transfer Speed Calculation

(68040 @ 20 MHz; 50 ns Clock Period):

37 ns max (BCLK to Data Out Valud of 68040) + 8 ns max (assumed 74xxx863 Port B to A delay) + 10 ns (assumed min data setup time to LIFE) = 55 ns

As can be seen from Equation #1, it is difficult to do zero wait state transfers (even during bursting) under worst case conditions, since the time required for one write transfer is greater than one clock period at 20 MHz. But, if the user is operating at a lower frequency (or with a different CPU that has better write data timing) and can perform a data transfer per bus clock, it is easy to implement by manipulating the GAL MC68040/LIFE interface equations shown later on in this application note.

RRD-B30M75/Printed in U. S. A.

III. LIFE LOCAL BUS PROTOCOL DESCRIPTION

The LIFE Local Bus Protocol employs the following signals:

CLK — The local bus clock input;

LCS — When the LIFE is a local bus slave this input signal indicates that the present local bus access is to the LIFE internal control and status registers (CSRs);

LASO — When the LIFE is local bus master this output signal indicates that an address will be valid at the next rising edge of CLK.

When the LIFE is local bus slave this signal indicates that the LIFE is available to do a data transfer operation, specifically that it is ready to accept an address from the host.

LASI — When the LIFE is local bus master this input signal indicates that the host will capture the LIFE address at the next rising edge of CLK, at which time the LIFE may continue on to the data transfer portion of the transaction.

When the LIFE is local bus slave this signal indicates that the host guarantees valid address at the next rising edge of CLK, at which time the LIFE may continue on to the data transfer portion of the transaction.

LDSO — When the LIFE is local bus master this output signal indicates that:

— Data will be valid at the next rising edge of CLK during a LIFE write access to the local bus;

— The LIFE is ready to input data during a read access from the local bus.

When the LIFE is local bus slave this signal indicates that the LIFE is available to do a data transfer operation, specifically that it is ready to read or write data from/to Futurebus+ (or its internal registers) for the host.

LDSI — When the LIFE is local bus master this input signal indicates that the host will read/write data from/to the Local Bus at the next rising edge of CLK.

When the LIFE is local bus slave this signal indicates that the host guarantees valid data at the next rising edge of CLK during a write to Futurebus+ or LIFE registers, or the host will read valid data at the next rising edge of CLK.

LReq — The LIFE asserts this output signal when it wishes to arbitrate for local bus mastership.

LGrant — This is an input to the LIFE from the Local Bus arbiter. When asserted it tells the LIFE that it is now the Local Bus Master.

CSRDec__ — This output signal is asserted by the LIFE when it has become the Local Bus Master in order to perform an access for some other Futurebus+ Master Module and the address (from the Futurebus+ Master Module) matches within the 4k byte CSR space of this module. This signal has traveled with the address through the Receive (or possibly Read) FIFO.

ExtMemDec__ — This output signal is asserted by the LIFE when it has become the Local Bus Master in order to perform an access for some other Futurebus+ Master Module and the address (from the Futurebus+ Master Module) matches within the 64-bit address and extent registers of the LIFE. This signal has traveled with the address through the Receive (or possibly Read) FIFO.

ExtUnitDec__ — This output signal is asserted by the LIFE when it has become the Local Bus Master in order to perform an access for some other Futurebus+ Master Module and the address (from the Futurebus+ Master Module) matches within the 64-bit address and extent registers of the LIFE. This signal has traveled with the address through the Receive (or possibly Read) FIFO.

LXABOutOE__ — This output signal controls the 74F543 transceiver Output Enables for both the Address and Byte Selects out of the LIFE to the Local Bus when the LIFE is the Local Bus Master.

LXAInOE__ — This output signal controls the 74F543 transceiver Output Enables for the Address coming into the LIFE from the Local Bus when the Host is the Local Bus Master.

LXAOutLE__ — This output signal controls the 74F543 transceiver Latch Enable for the Address coming out of the LIFE to the Local Bus when the LIFE is the Local Bus Master. This address will be latched for the duration of the data transfer cycle.

LXBInOE__ — This output signal controls the 74F543 transceiver Output Enables for the Byte Enables coming into the LIFE from the Local Bus when the Host is the Local Bus Master.

LXBOutLE__ — This output signal controls the 74F543 transceiver Latch Enable for the Byte Selects coming out of the LIFE to the Local Bus when the LIFE is the Local Bus Master. The Byte Selects will be latched for the duration of the data transfer cycle.

LXDOutOE__ — This output signal controls the 74XXX863 transceiver Output Enables for the Data out of the LIFE to the Local Bus when the LIFE is outputting Data to the Local Bus.

LXDInOE__ — This output signal controls the 74XXX863 transceiver Output Enables for Data coming into the LIFE from the Local Bus when the Host is inputting data to the LIFE.

LErr__ — This input/output signal is asserted by the LIFE when it has become the Local Bus Master in order to perform an access for some other Futurebus+ Master Module and there was some kind of error during the address or data portion of the transfer (Ex. Parity Error).

	It may also be asserted when the LIFE is a local bus slave to the 68040 (''LParEn__'' input asserted) and input parity does not agree with LIFE generated parity.
LParEn__	This input when asserted tells the LIFE to check the Local Bus input parity (Address, Byte Selects, Command, and Data), if a parity error exists the ''LErr__'' I/O pin will be asserted.
LSysReset__	This I/O signal allows the local bus to force a reset of all Futurebus+ interface logic but does not drive the Freo output to Futurebus+. A Futurebus+ FREf assertion will assert LSysReset to force a system reset of the module. The module will respond with LResetDone when the reset is completed, this was not implemented in the current simulations.
LResetDone	Not implemented in these simulations.
LInterrupt__	Interrupt output to the Local Bus.
LAD[31:0]	The multiplexed Address and Data I/O of the LIFE.
LBP[3:0]	The Byte Parity of the Address and Data I/O of the LIFE.
LCM[7:0]	The Command I/O of the LIFE.
LCP	The Command Parity I/O bit of the LIFE.

This Local Bus interface supports the following types of operations (also see the Local Bus Timing Figures):

— Read and Write (single and burst) transactions across Futurebus+ with the LIFE acting as the Local Bus Master and a Futurebus+ Slave Module;

— Read and Write (single and burst) transactions across Futurebus+ with the Host acting as the Local Bus Master and the LIFE becoming a Futurebus+ Master Module;

— Read and Write transactions to the LIFE internal registers with the Host acting as the Local Bus Master.

— Read and Write transactions to the LIFE internal registers with the LIFE acting as the Local Bus Master (in response to a Futurebus+ transaction). These types of transactions are not implemented in these simulations of the LIFE.

The LIFE generates parity for all Local Bus inputs to the FIFOs (Address, Command, Byte Selects, and Data). That is why in this system simulation parity is not generated on the local bus at all and the ''LParEn__'' input is negated (LIFE does not check input parity). If the system designer has parity generated on the Local Bus he will need an extra 74F543 latched transceiver for the address parity bits and the command parity bit, they can both go in the same transceiver. The LIFE will perform a parity check of all inputs if the ''LParEn__'' input is asserted.

During a Host Read Access from Futurebus+ the LIFE Local Bus Interface will take in the Address for the initial Read access into the Transmit FIFO then force a bus relinquish and retry of the transaction by negating LASO and LDSO. Once the transaction reaches the head of the Transmit FIFO the LIFE will arbitrate for Futurebus+ (it supports both Distributed and Central Arbitration Schemes) and complete the Read transaction. Note that this Read transaction may be split by the slave if the Master and Slave are capable of supporting Split transactions (the LIFE can support one outstanding Split Read Transaction). The LIFE Local Bus Interface will continue to force transaction retries to the Host until read data is returned via Futurebus+ (or some type of error condition occurs and asserts LInterrupt), the interface will then allow the transaction to complete. A Host Read Access from Futurebus+ can be seen at the end of the Local Bus Timing Figures.

During a Host Write Access the Address, Data, Byte Selects, Command and Parity Bits are enqueued into the Transmit FIFO. Once all the Data is enqueued the Local Bus Transaction will complete. As far as the Host is concerned the Data has been written to its Futurebus+ destination. Once the transaction reaches the head of the Transmit FIFO the LIFE will arbitrate for Futurebus+ (it supports both Distributed and Central Arbitration Schemes) and complete the write transaction. If some error occurs LInterrupt will be asserted.

When another Futurebus+ Master selects this module to do a Read/Write access from/to the Local Bus of this module the transaction gets enqueued into the Receive FIFO. When it reaches the head of the Receive FIFO the LIFE arbitrates (LReq output asserted) for the local bus and once granted (LGrant input asserted) asserts the Address, Command, Parity bits and LASO. Once the Host asserts LASI the LIFE asserts the Byte Selects, Parity bits and LDSO. Once LDSI is asserted the data transfer can begin using the LDSO, LDSI protocol. When the transfer completes both LDSO and LASO are negated.

Another issue is that of a Futurebus+ read or write of the LIFE CSR space. This can happen when the LIFE is acting as a Futurebus+ Slave Module and a Master Module Reads/Writes from/to its CSR space. This transaction will be treated as a normal Futurebus+ induced Read/Write access and the address, and data during a write access, will be enqueued into the Receive FIFO of the LIFE along with the address decode (CSRDec__ asserted in this case). When this reaches the head of the Receive FIFO the LIFE will arbitrate for the Local Bus (LReq output asserted). Once it becomes master of the Local Bus (LGrant input asserted) it will output the address, CSRDec__, Command and Byte Selects along with LASO asserted. If the user wishes the CSR access to access the internal LIFE registers he simply asserts the LSC__ input and the access will take place within the LIFE. The user may optionally access external RAM/ROM CSR. The data will be transferred via the LDSO, LDSI protocol. This method gives complete flexibility to the user as to having any combination of LIFE internal CSR or external RAM/ROM CSR accesses.

IV. PROGRAMMABLE LOGIC DESCRIPTION

The GAL (or similar type PAL) programmable logic devices are straightforward to understand based upon looking at the equations that follow and the local bus timing. Even though they are simple to understand I have included a brief description of each GAL as follows:

1. 'pal__68040'

   This GAL is the most complex of the GALs as it performs the interface between the MC68040 CPU /the local bus/ and the LIFE Futurebus+ Protocol and Data Path Unit.

In other words it translates LIFE Local Bus Master protocol into 68040 protocol, and it translates Host (68040) protocol into LIFE protocol. Most of the terms are self explanatory except:

ADEL__ This clocked delay term asserts on the rising clock edge after the LIFE LASO asserts (address phase begins) and negates once the term CYCLING__ asserts (data phase begins);

CYCLING__ This clocked delay term asserts on the rising clock edge after the LIFE LDSO asserts (data phase begins) and negates once the access is over (TS__ asserted or LASO, TIP__ negated);

DATDEL__ This clocked delay term asserts on the rising clock edge after the LIFE LDSO asserts (data phase begins) and toggles during the data phase to guarantee two clocks per data transfer. This term negates for the final time once the access is over (TS__asserted or LASO negated);

2. 'cmd__pal'

   This GAL converts 68040 control lines (partial, locked, and Read/Write accesses) into Futurebus+ compatible address phase commands and Futurebus+ commands into local bus control lines.

3. 'arb'

   This GAL provides the Local Bus Arbitration mechanism between the MC68040 (BR__, BG__, and BB__) and the LIFE Futurebus+ Protocol Controller (LReq and LGrant). It also provides Local Bus to Futurebus+ command translation for CM5, 6, 7 and MC68040 address decoding for Futurebus+ (CSFBUS__) or the LIFE internal registers (CSREG__).

```
begin header
  Company:      National Semiconductor Corp.
  Author:       Rusty Meier
  Date:         May 20, 1993
  Revision:     1.0
  Description:  This PAL performs the interface between the
                68040 CPU and the Logical Interface Futurebus+
                Engine (LIFE) in both the Master
                and Slave mode.  This interface allows the
                68040 to read or write to the LIFE
                Registers and/or Futurebus+ using the LIFE
                FIFOs.  The interface also allows the LIFE
                to become Master of the local bus and perform
                read and write accesses.
                This interface allows 2 clock accesses from
                68040 and for the LIFE.
  Module Name:  'pal_68040'
end header

begin Definition
DEVICE g22v10;

Inputs
        CLK =1,         TS_ =2,         TIP_ =3,
        RW_ =4,         BGI_ =5,        LASO =6,
        LDSO =7,        LGrant = 8,     DTACK_ = 9,
        Partial =10,    LBCS_ =11,      FBCS_ =13;

Output (COM)
        LDSI =21,       !TA_ =18,       CM2 =15;

Output (REG)
        !OTIP_ =16;

Feedbacks (REG)
        !CYCLING_ =23,  !ADEL_ =20,     !DATDEL_=19,
        DATD2 = 14;

Feedbacks (COM)
        LASI = 22,      !TEA_ =17;

end Definition

begin Equations

CYCLING_ := !LBCS_ & !LGrant & LASO &
                    LDSO & TS_ & ADEL_    {CPU accessing}
     # CYCLING_ & !LGrant & LASO & LDSO &
                    !TIP_ & TS_           {Hld CYCLING_}
     # CYCLING_ & !LGrant & LDSO &
                    !TIP_ & TS_           {Hld CYCLING_}
     # LGrant & LASO & LDSO;              {LIFE accessing}

LASI = !LBCS_ & !LGrant & LASO & !TIP_
          & !TS_ & !CYCLING_ & !TEA_      {CPU add ready}
     # !LBCS_ & !LGrant & LASO & !TIP_
          & ADEL_ & !CYCLING_ & !TEA_     {CPU add ready}
     # LASI     & !LGrant & LASO & LDSO & !TIP_
                    & TS_ & !TEA_         {Hld CPU LASI}
     # LASI     & !LGrant & LDSO & !TIP_
                    & TS_ & !TEA_         {Hld CPU LASI}
     # LGrant   &  BGI_ & LASO;           {LIFE add ready}


LDSI = !LBCS_ & !LGrant & LDSO & !TIP_ & TS_
```

TL/F/11484–1

```
        & ADEL_ & !DATDEL_ & Partial & !TEA_ {CPU Partial}
     # !LBCS_ & !LGrant & LDSO &
           !TIP_ & TS_ & DATD2 & RW_       {CPU Read}
     # !LBCS_ & !LGrant & LDSO &
           !TIP_ & TS_ & DATDEL_ & !RW_    {CPU Write}
     # LGrant   & BGI_ & LASO & LDSO &
                  ADEL_  & !DATDEL_    {LIFE Partial}
     # LGrant   & BGI_ & LASO & LDSO &
                  DATDEL_ & !DTACK_;  {LIFE data RDY}

ADEL_ := !LBCS_ & !LGrant & !TS_ & CYCLING_ {CPU add delay}
       # !LBCS_ & !LGrant & LASO & !CYCLING_
              & !DATDEL_ & !LDSO & !TIP_   {CPU add delay}
       # ADEL_ & !LGrant & LASO & !CYCLING_
          & !DATDEL_ & !LDSO & TS_ & !TIP_  {Hold ADEL_}
       # LGrant  & BGI_ & LASO & !CYCLING_
                  & !DATDEL_ & !LDSO;  {LIFE add delay}

DATDEL_ := !LBCS_ & !LGrant & LDSO & TS_
           & ADEL_ & !DATDEL_ & !DATD2     {CPU Data #1 Del}
         # !LBCS_ & !LGrant & LDSO & TS_
           & CYCLING_ & !DATDEL_ & !DATD2  {CPU Data #1 Del}
         # BGI_ & LGrant & LASO & LDSO
                          & !DATDEL_; {LIFE Data Delay}

TA_  = !LBCS_ & !LGrant & LDSO &
                !RW_ & !TEA_ & DATDEL_  {CPU Write Ready}
     # !LBCS_ & !LGrant & LDSO &
                 RW_ & !TEA_ & DATD2    {CPU Read  Ready}
     # !LBCS_ & !LGrant & !TIP_ & CYCLING_
        & LASI & TS_ & !LASO & !LDSO     {CPU Bus Backoff/Retry}
     # !LBCS_ & !LGrant & !TIP_ & ADEL_
                  & !LASO & !LDSO;    {CPU Bus Backoff/Retry}

TEA_ = !LBCS_ & !LGrant & !TIP_ & CYCLING_
        & LASI & TS_ & !LASO & !LDSO     {CPU Bus Backoff/Retry}
     # !LBCS_ & !LGrant & !TIP_ & ADEL_
                  & !LASO & !LDSO;    {CPU Bus Backoff/Retry}

OTIP_ := LGrant & BGI_ & LASO & LASI;    {LIFE gen TIP_}

CM2 = !LGrant & LGrant;                  {Response, make=0}

DATD2 := !LBCS_ & !LGrant & LASO & LDSO  {Caching,CM3, make=0}
           & DATDEL_ & TS_ & !DATD2      {CPU Data #2 Del}
         # BGI_ & LGrant & LASO & LDSO
           & DATDEL_ & !DATD2;           {LIFE Data2 Delay}


[CYCLING_,ADEL_,DATDEL_,OTIP_].c = CLK;
OTIP_.oe   = LGrant & BGI_;
[TA_,TEA_].oe = !LGrant;
[DATD2,CM2].oe = !LGrant & !TIP_;

end Equations
```

TL/F/11484–2

```
begin header
   Company:     National Semiconductor Corp.
   Author:      Rusty Meier
   Date:        October 15, 1992
   Revision:    0.5
   Description: This PAL handles the arbitration between
                the 68040 and the LIFE for the local
                bus. It does this through the LReq & LGrant
                I/O of the LIFE and the Bus Request (BR_),
                Bus Grant (BG_) and Bus Busy (BB_) I/O of
                the 68040.
                This PAL also generates command bits CM<7,6,5,3>
                for 68040 Futurebus+/LIFE accesses.
   Module Name: 'arb'
end header

begin definition
Device gal16v8;

Inputs
        CLK=1,          LReq=2,         BR_=3,
        BB_=4,          A31=5,          A30=6,
        TIP_=7,         TA_=8,          TEA_=9;

Outputs (COM)
        !CSREG_=19,     !CSFBUS_=18,    CM5=14,
        CM6=13,         CM7=12;

Feedbacks (REG)
        LGrant=17,      !BGI_=16;

Feedbacks (COM)
        !BG_=15;

end definition

begin equations

CSREG_ =  !LGrant & A31 & A30 & !TIP_; {CPU accessing LIFE Reg}

CSFBUS_ = !LGrant & !A31 & !A30 & !TIP_; {CPU accessing FBUS/32}

LGrant := !BG_ & BB_ & LReq    {CPU not GRNTed, LIFE RQST}
        #  BR_ & BB_ & LReq;   {CPU no RQST, LIFE RQST}

BGI_ := !BR_ & !LReq                  {CPU RQST, LIFE no RQST}
      # !BR_ & BG_                    {Hold while RQST & GRNT}
      # !BR_ & !LReq & TA_ & !TIP_
      # !BR_ & BG_   & TA_ & !TIP_
      # !BR_ & !LReq & TEA_ & !TIP_
      # !BR_ & BG_   & TEA_ & !TIP_;

[LGrant,BGI_].c = CLK;

BG_ = BGI_ & TA_           {68040 Bus Grant}
    # BGI_ & TEA_;

CM5 = BG_ & !BG_;           {32 bit address}

CM6 = BG_ & !BG_;                  {CM6 = 0}

CM7 = BG_ & !BG_;                  {CM7 = 0}

[CM5,CM6,CM7].oe = !LGrant & !TIP_;

end equations
```

TL/F/11484–4

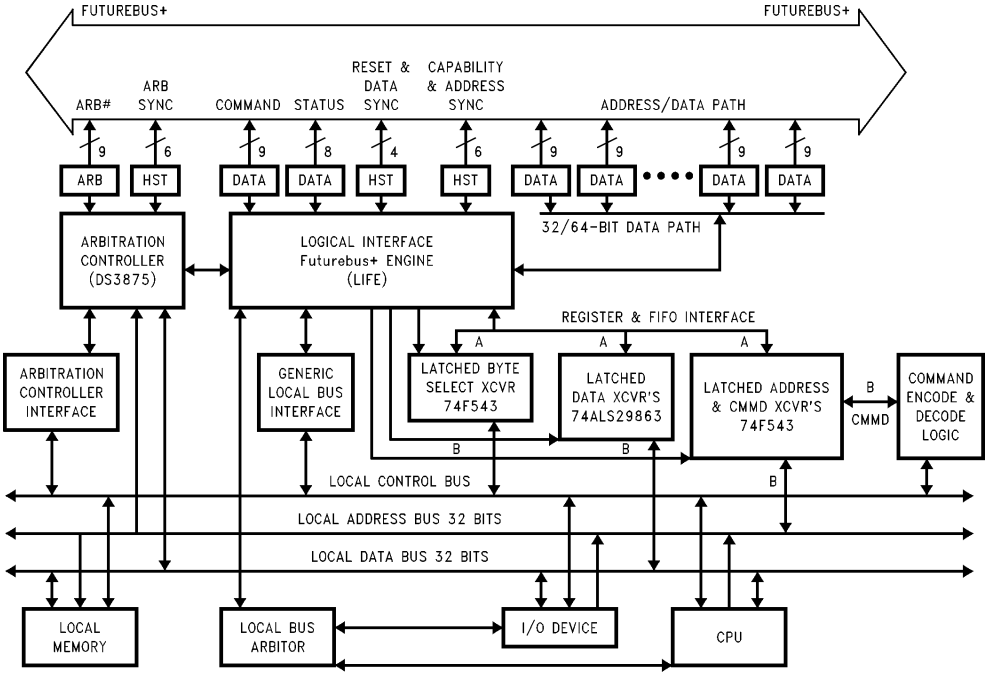
```
begin header
  Company:       National Semiconductor Corp.
  Author:        Rusty Meier
  Date:          January 7, 1993
  Revision:      0.3
  Description:   This PAL translates the local bus control
                 values to the Futurebus+ command lines
                 CM (7:0) and from Futurebus+ command to
                 local bus control, depending upon whether
                 the local bus is a master or slave to the
                 LIFE Protocol Controller.
  Module Name:   'cmd_pal'
end header

begin Definition
DEVICE gal16v8;

inputs
        LGrant =1,      TS_ =2,         TIP_ =3,
        SIZ0 =4,        SIZ1 =5,        A0 =6,
        A1 =7,          LOCK_ =8,       LOCKE_ =9,
        LASO = 11;

outputs (com)
        CM1 = 19,       CM0 = 12;

feedbacks (COM)
        !BE3_ = 18,     !BE2_ = 17,
        !BE1_ = 16,     !BE0_ = 15,
        CM4 = 14,       RW_ = 13;

end Definition

begin Equations

BE3_ = !TS_  & !SIZ1 &  SIZ0 &  A1 &  A0   {assert BE3_}
     # !TS_  &  SIZ1 & !SIZ0 &  A1 & !A0   {assert BE3_}
     # !TS_  & !SIZ1 & !SIZ0               {assert BE3_}
     # !TS_  &  SIZ1 &  SIZ0               {assert BE3_}
     # BE3_  & !TIP_;

BE2_ = !TS_  & !SIZ1 &  SIZ0 &  A1 & !A0   {assert BE2_}
     # !TS_  &  SIZ1 & !SIZ0 &  A1 & !A0   {assert BE2_}
     # !TS_  & !SIZ1 & !SIZ0               {assert BE2_}
     # !TS_  &  SIZ1 &  SIZ0               {assert BE2_}
     # BE2_  & !TIP_;

BE1_ = !TS_  & !SIZ1 &  SIZ0 & !A1 &  A0   {assert BE1_}
     # !TS_  &  SIZ1 & !SIZ0 & !A1 & !A0   {assert BE1_}
     # !TS_  & !SIZ1 & !SIZ0               {assert BE1_}
     # !TS_  &  SIZ1 &  SIZ0               {assert BE1_}
     # BE1_  & !TIP_;

BE0_ = !TS_  & !SIZ1 &  SIZ0 & !A1 & !A0   {assert BE0_}
     # !TS_  &  SIZ1 & !SIZ0 & !A1 & !A0   {assert BE0_}
     # !TS_  & !SIZ1 & !SIZ0               {assert BE0_}
     # !TS_  &  SIZ1 &  SIZ0               {assert BE0_}
     # BE0_  & !TIP_;
[BE3_,BE2_,BE1_,BE0_].oe = !LGrant;

CM4 = !LGrant & !TS_  & !RW_   {Assert when write}
    # !LGrant & !TIP_ & !RW_;  {Assert when write}

CM1  = !LGrant & SIZ1 & !SIZ0 & !TS_   {Partial Access}
     # !LGrant & SIZ0 & !SIZ1 & !TS_   {CPU Partial}
     # !LGrant & SIZ1 & !SIZ0 & !TIP_  {CPU Partial}
```

TL/F/11484–5

```
     # !LGrant & SIZ0 & !SIZ1 & !TIP_; {CPU Partial}

CM0  = !LGrant & !LOCK_ & LOCKE_;   {CPU Locked Access}
[CM4,CM1,CM0].oe = !LGrant & !TIP_;

RW_ = LGrant & !CM4 & LASO;       {LIFE Read Access}
RW_.oe = LGrant & LASO;

end Equations
```

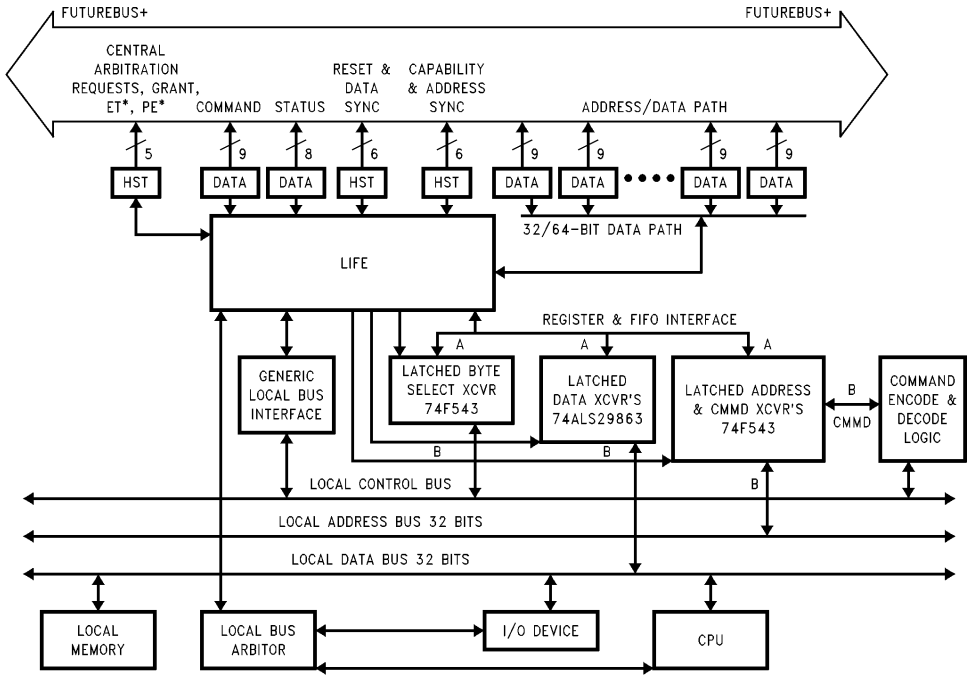
TL/F/11484–6

NSC's Futurebus+ Chipset Module Block Diagram Using Distributed Arbitration



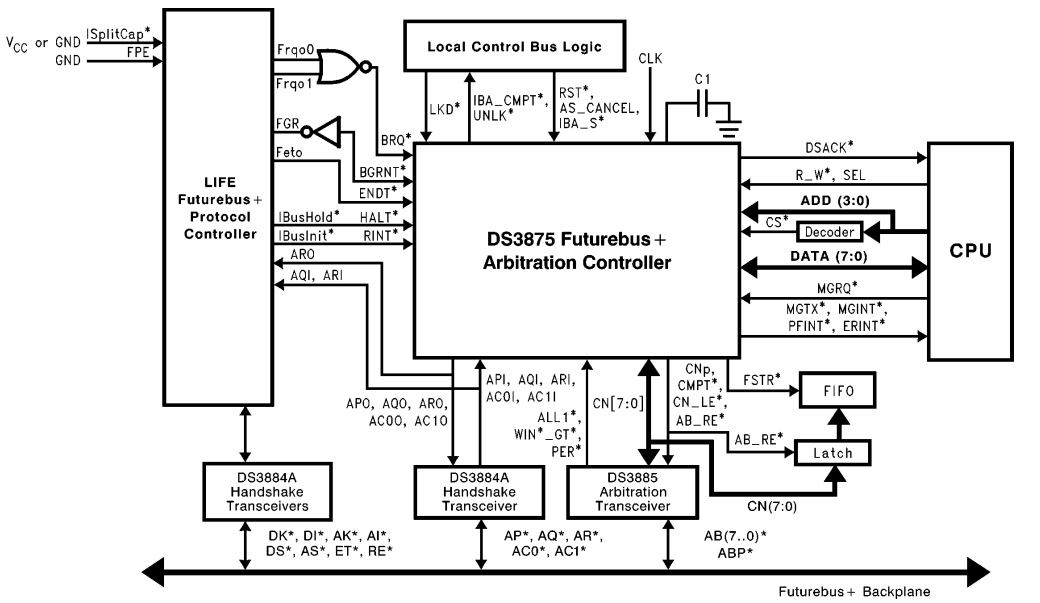
TL/F/11484–7

NSC's Futurebus+ Chipset Module Block Diagram Using Central Arbitration

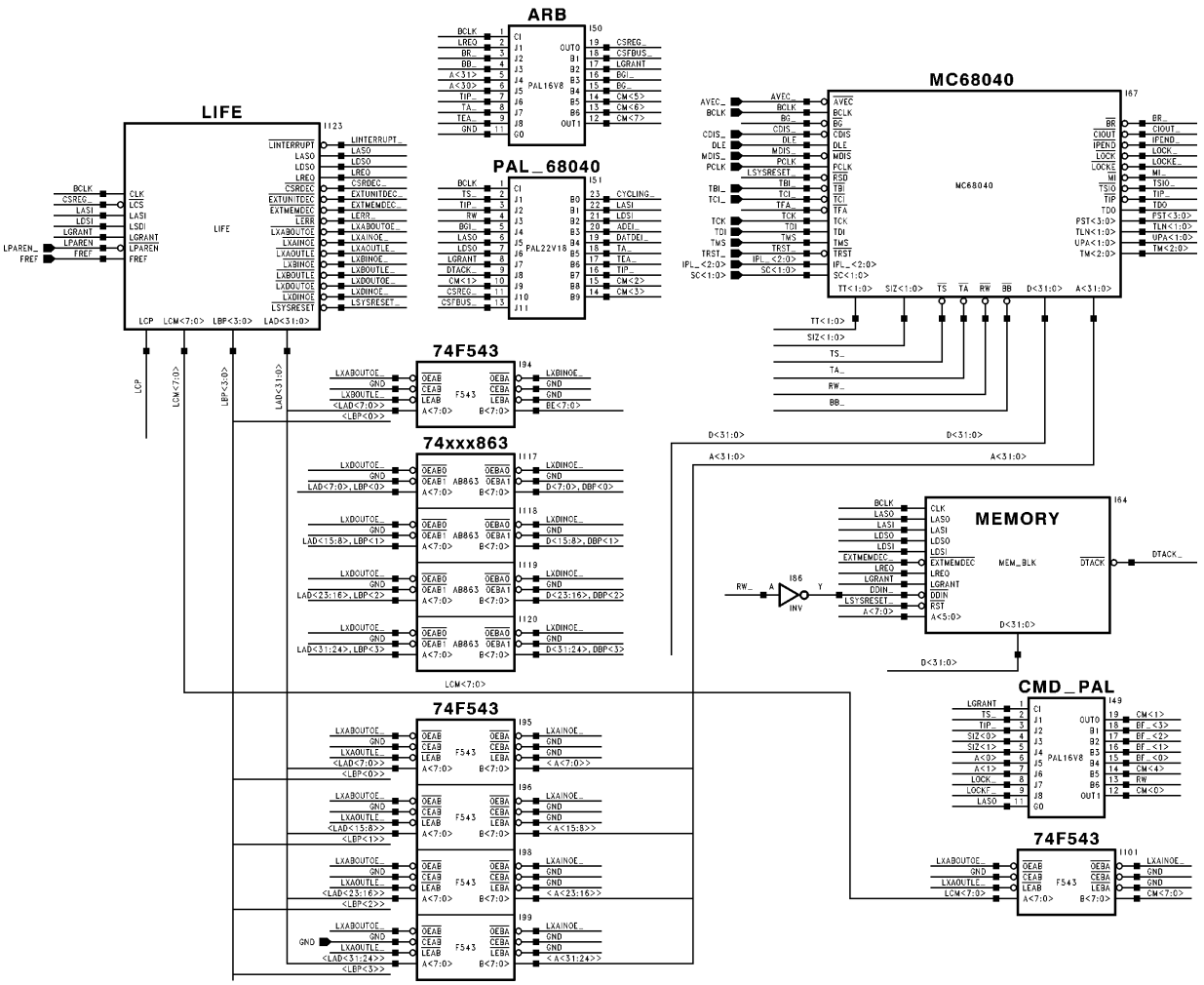


TL/F/11484–8

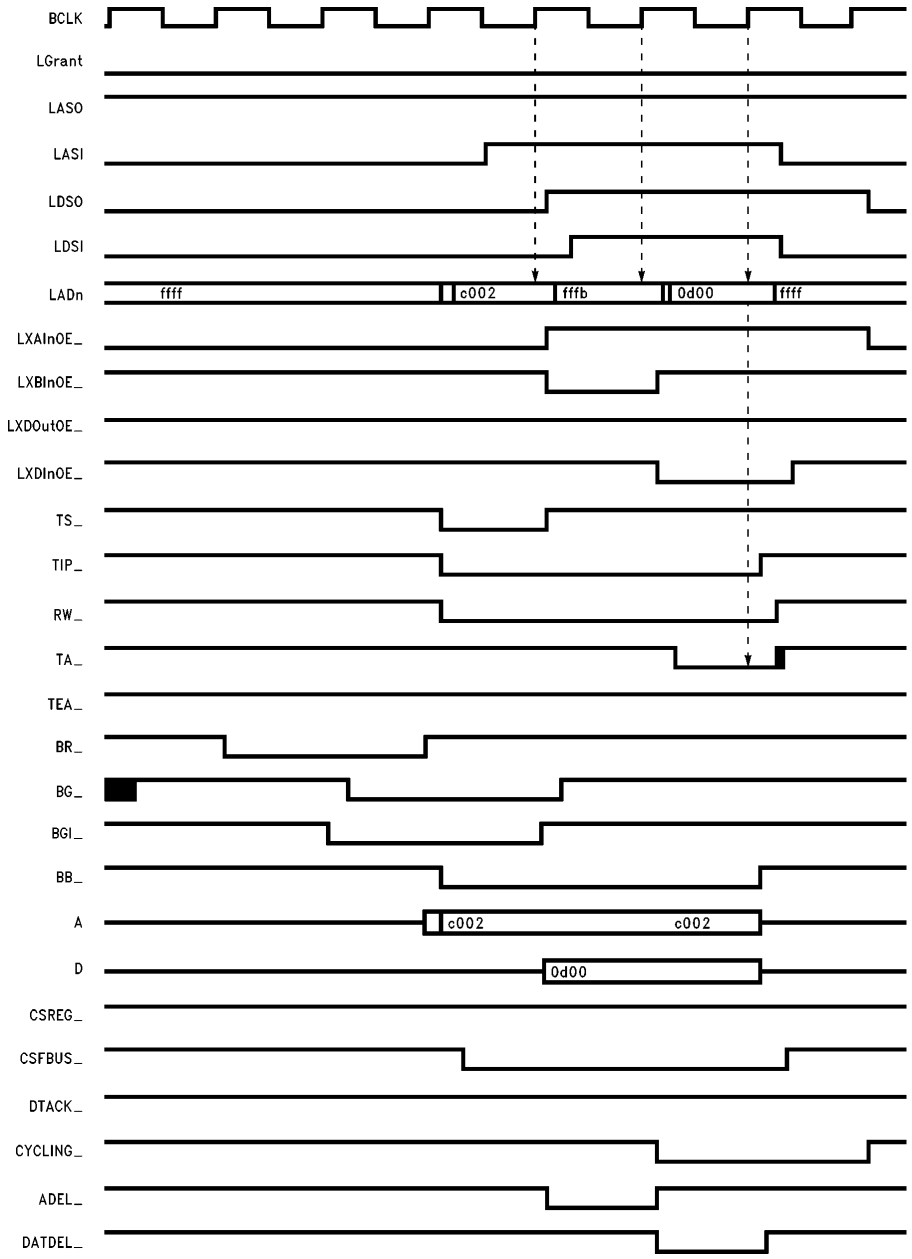
LIFE to DS3875 Interface



TL/F/11484–9

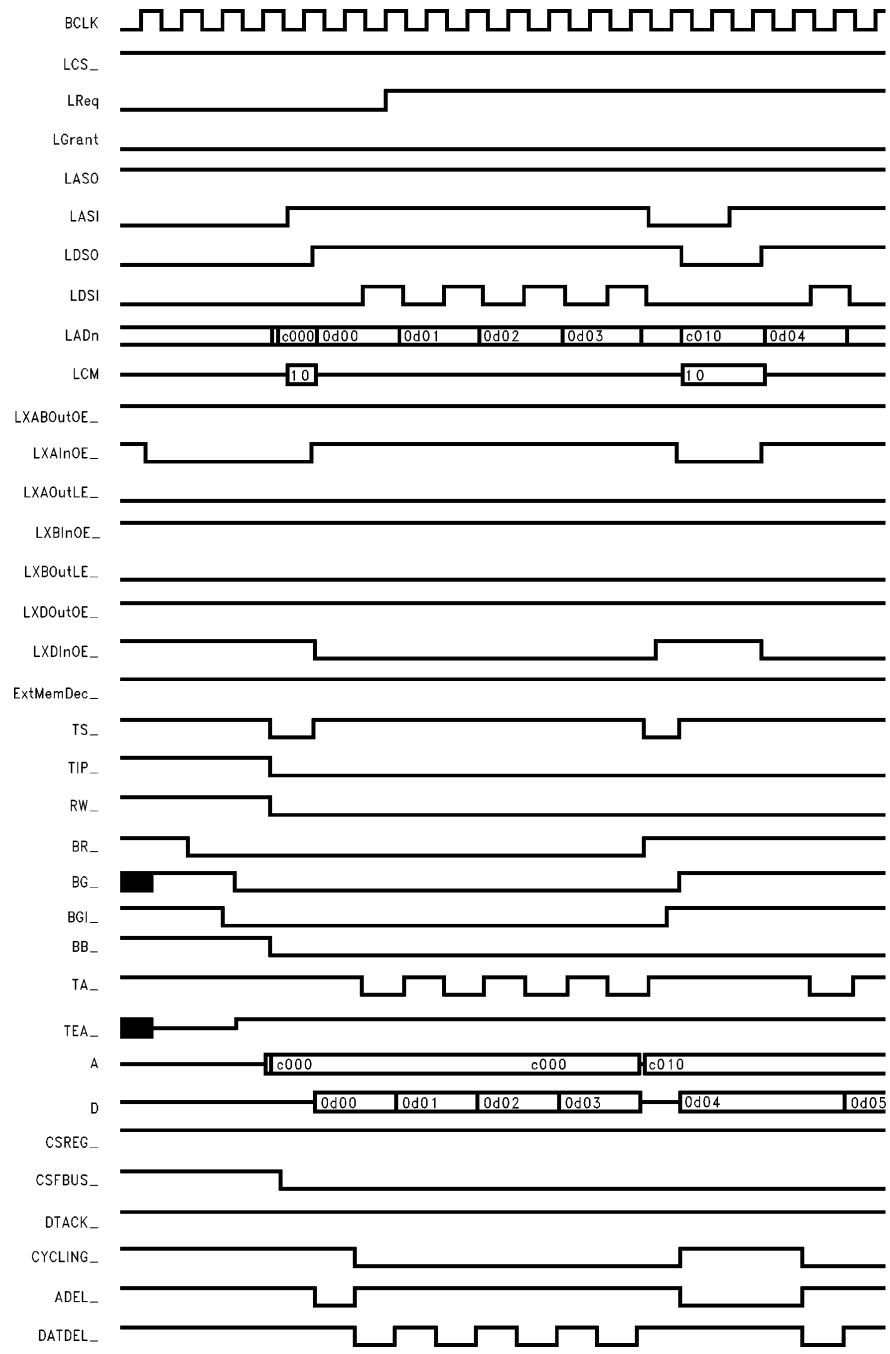

68040/LIFE_Simulation
ARB
PAL_68040
LIFE
MC68040
74F543
74xxx863
MEMORY
CMD_PAL
74F543
74F543
TL/F/11484–10

68040 Partial Write Access to Futurebus+ (thru LIFE)



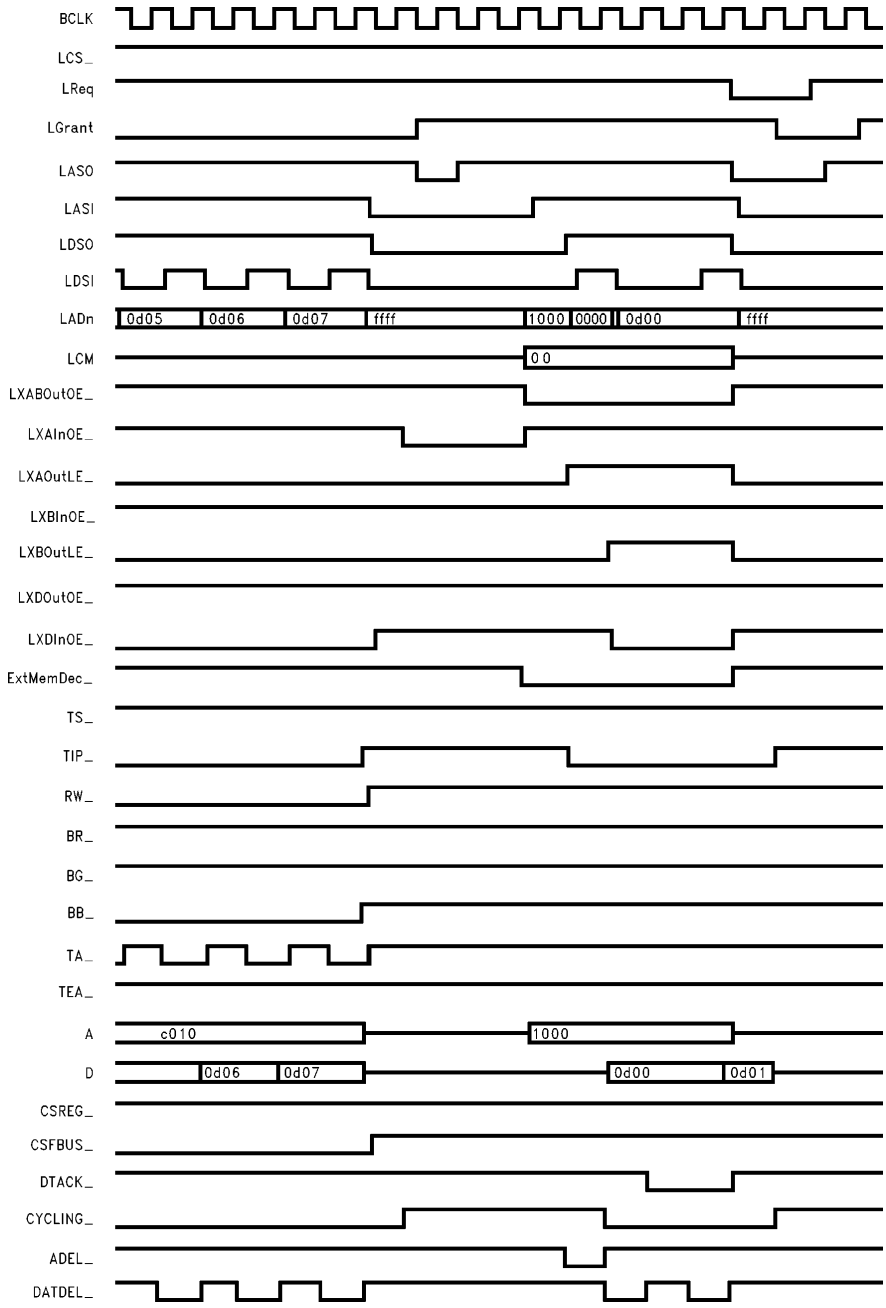
TL/F/11484–11

68040 Burst Write to Futurebus+ (thru LIFE)

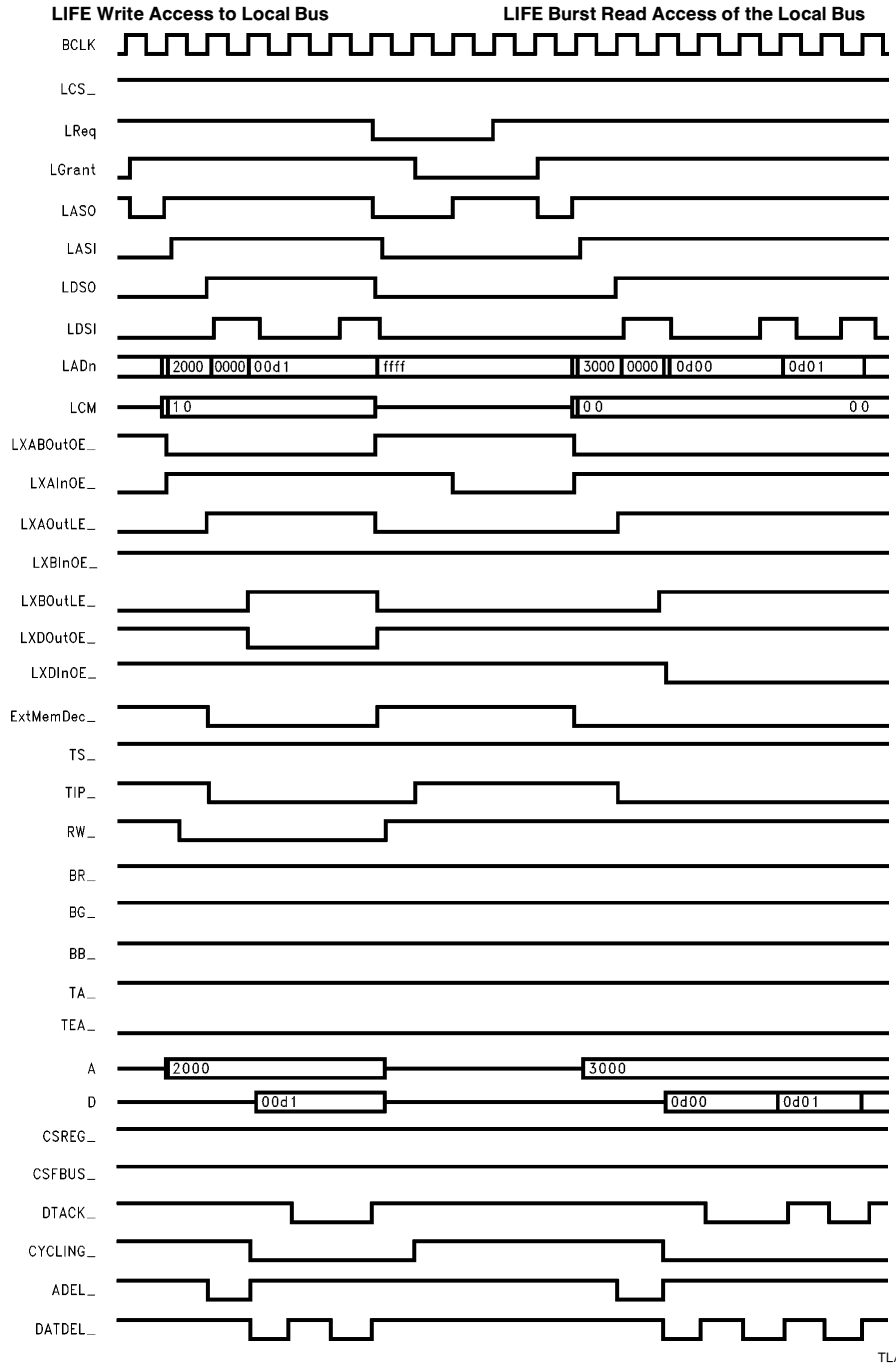


TL/F/11484–12

68040 Burst Write to Futurebus+ (thru LIFE) (Continued) **LIFE Read Access from Local Bus**

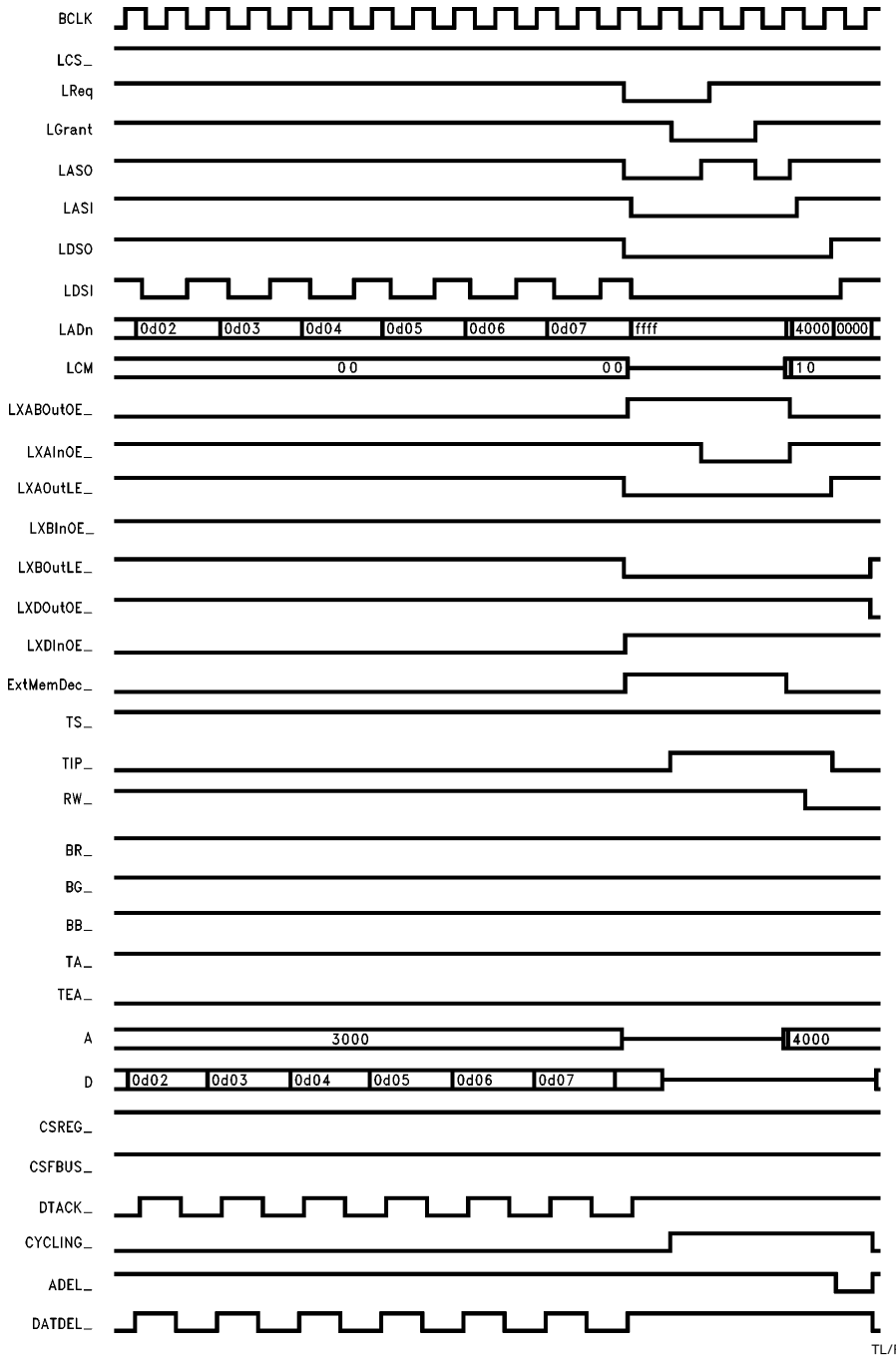


TL/F/11484–13


LIFE Write Access to Local Bus
LIFE Burst Read Access of the Local Bus
BCLK
LCS_
LReq
LGrant
LASO
LASI
LDSO
LDSI
LADn
2000
0000
00d1
ffff
3000
0000
0d00
0d01
LCM
10
00
00
LXABOutOE_
LXAInOE_
LXAOutLE_
LXBInOE_
LXBOutLE_
LXDOutOE_
LXDInOE_
ExtMemDec_
TS_
TIP_
RW_
BR_
BG_
BB_
TA_
TEA_
A
2000
3000
D
00d1
0d00
0d01
CSREG_
CSFBUS_
DTACK_
CYCLING_
ADEL_
DATDEL_

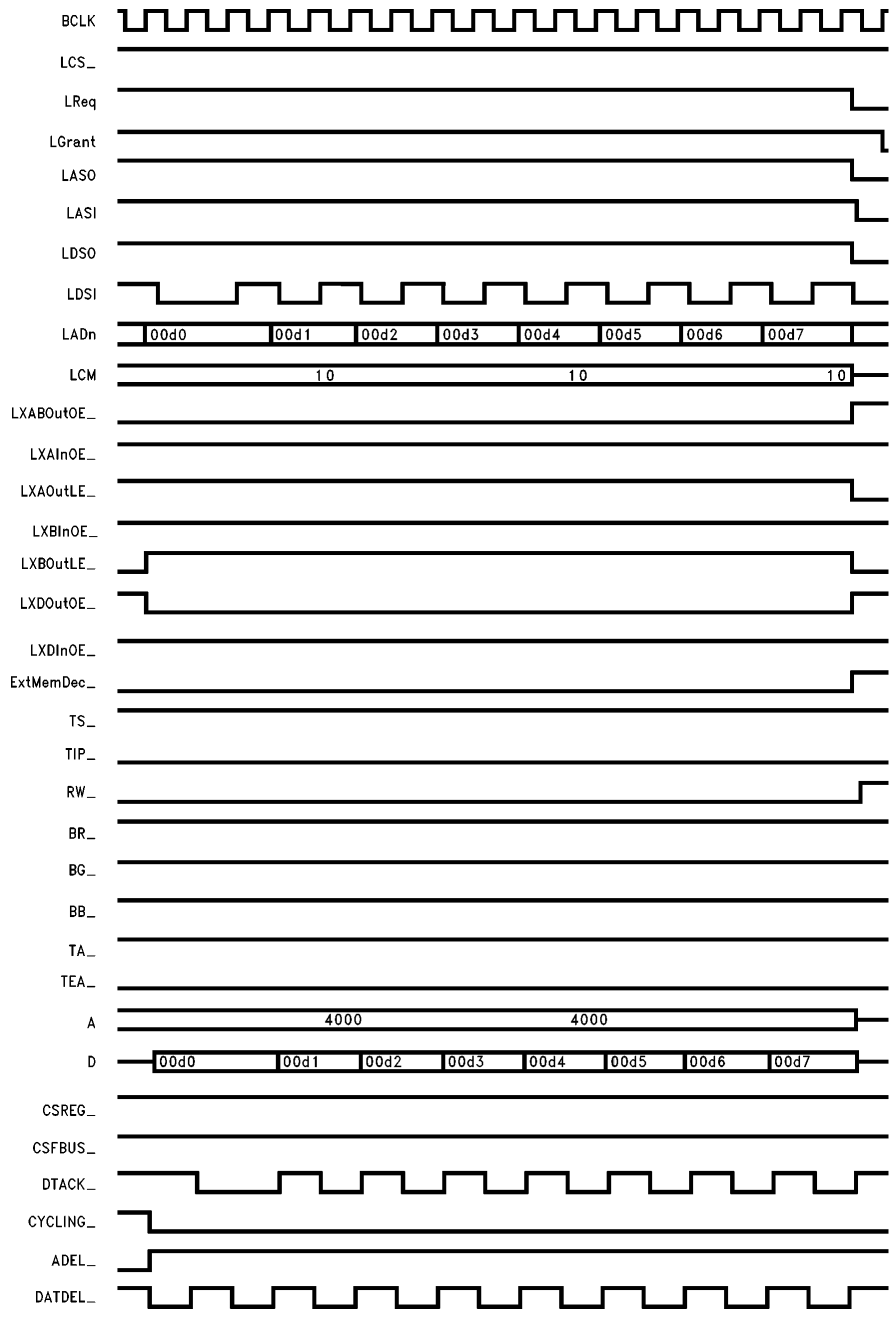
TL/F/11484–14

LIFE Write Access to Local Bus (Continued) **LIFE Burst Read Access of the Local Bus**

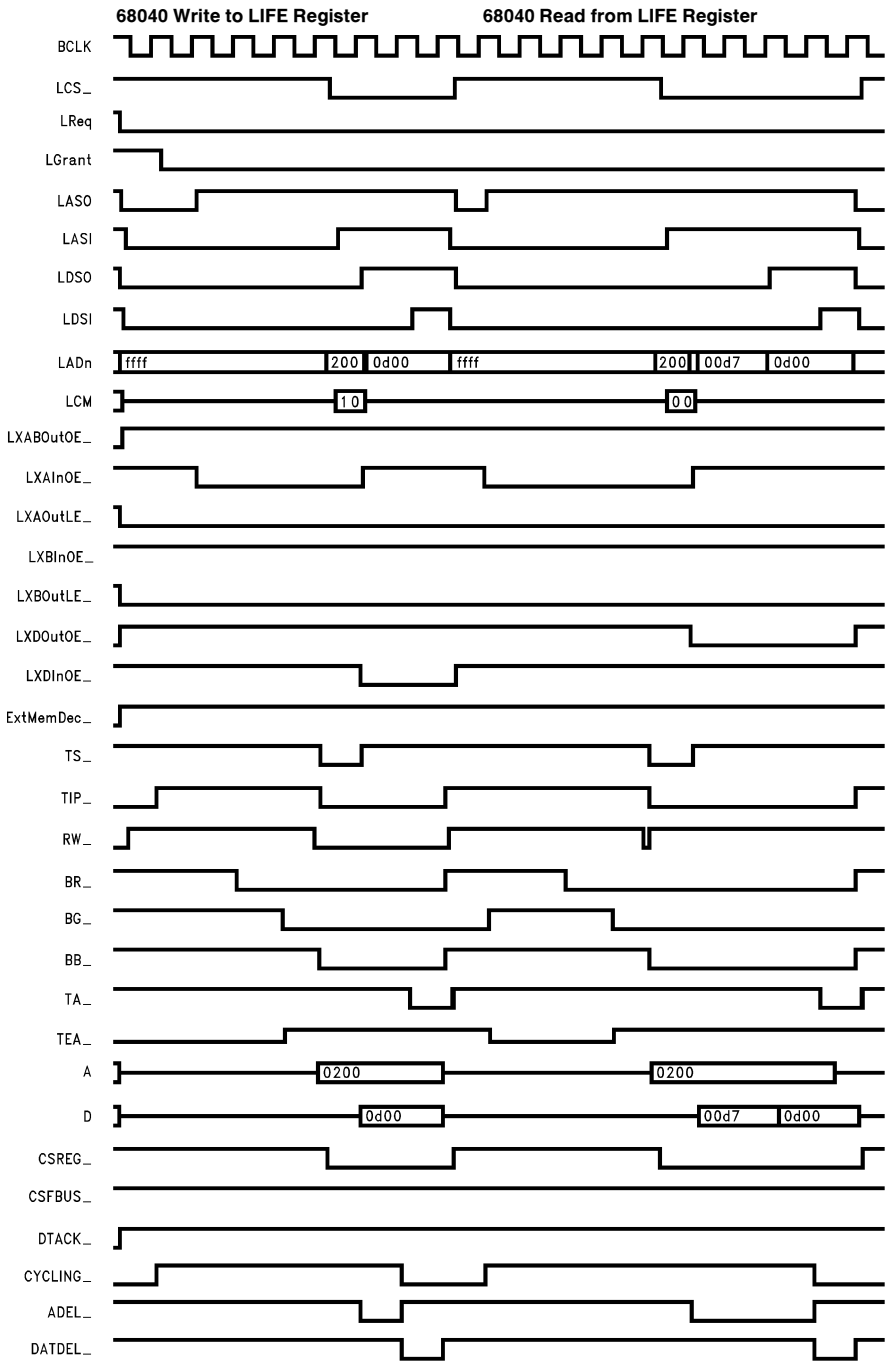


TL/F/11484–15

LIFE Burst Write Access to Local Bus (Continued)

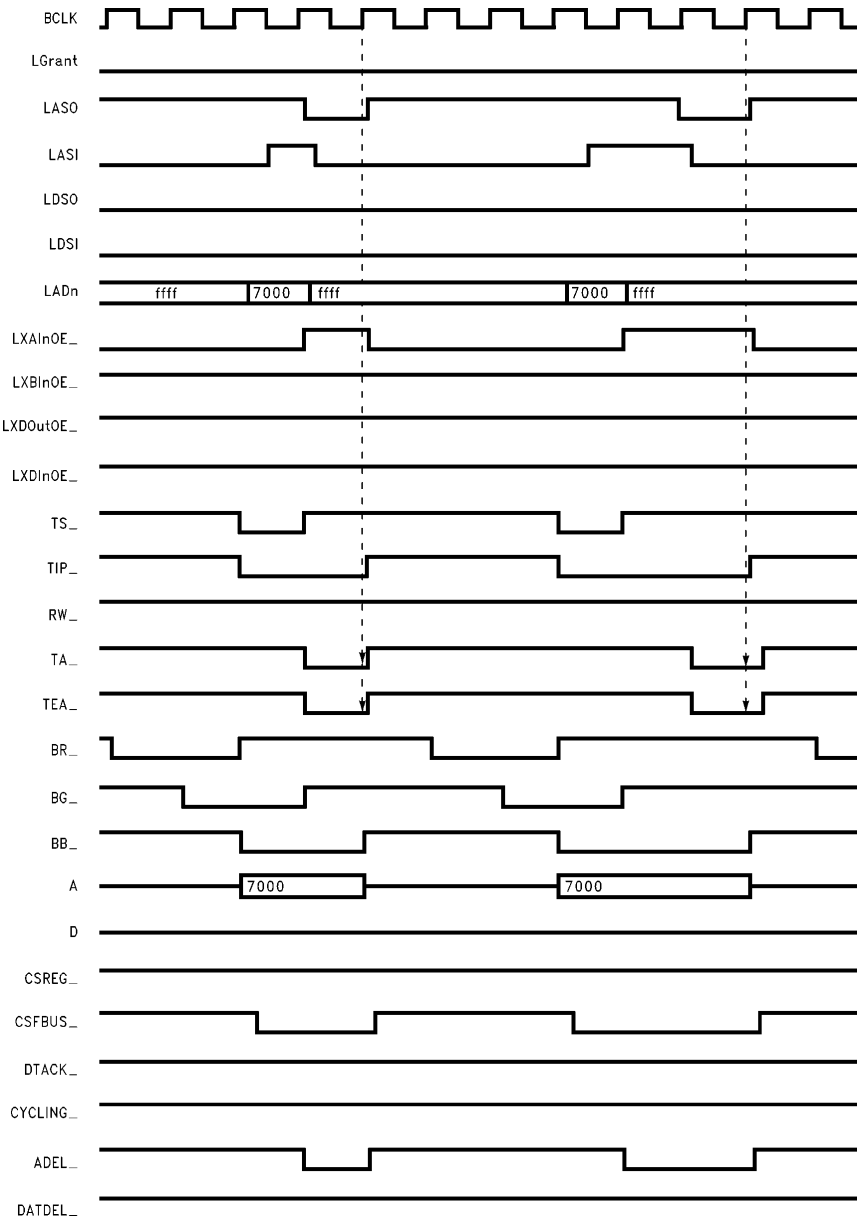


TL/F/11484–16


68040 Write to LIFE Register
68040 Read from LIFE Register
BCLK
LCS_
LReq
LGrant
LASO
LASI
LDSO
LDSI
LADn
ffff
200
0d00
ffff
200
00d7
0d00
LCM
10
00
LXABOutOE_
LXAInOE_
LXAOutLE_
LXBInOE_
LXBOutLE_
LXDOutOE_
LXDInOE_
ExtMemDec_
TS_
TIP_
RW_
BR_
BG_
BB_
TA_
TEA_
A
0200
0200
D
0d00
00d7
0d00
CSREG_
CSFBUS_
DTACK_
CYCLING_
ADEL_
DATDEL_

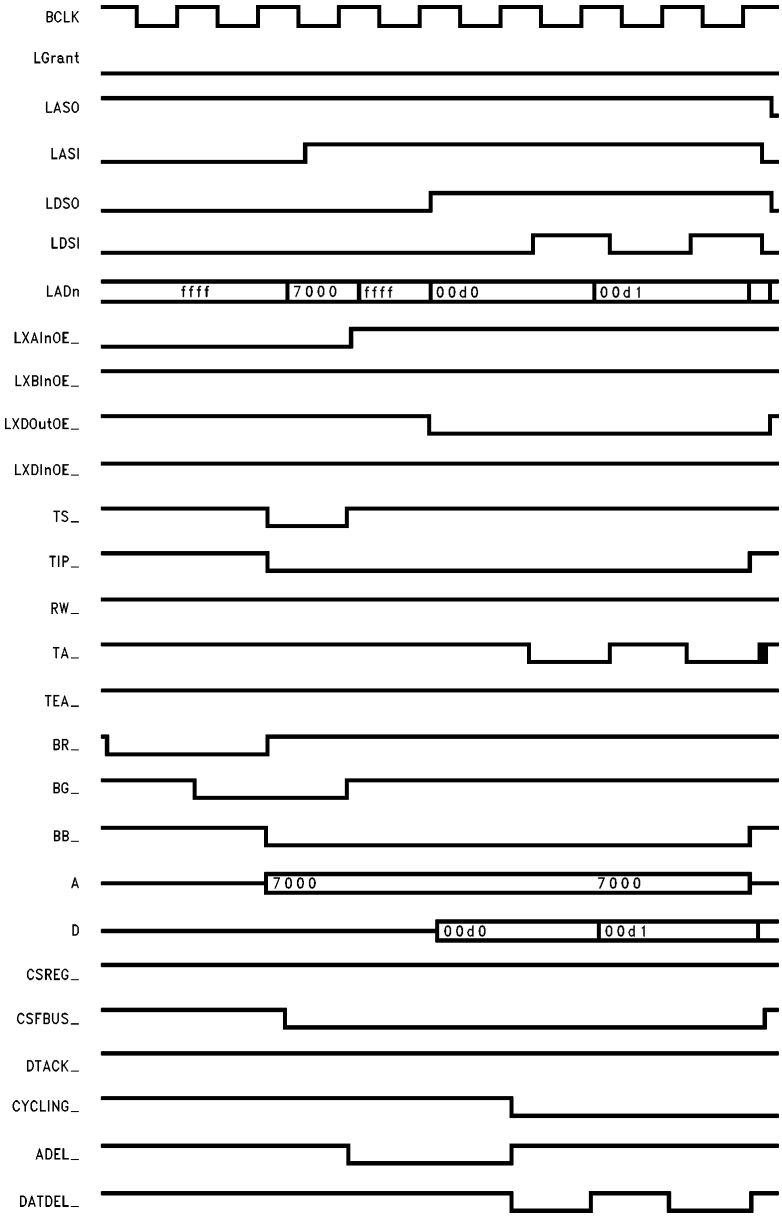
TL/F/11484–17

68040 Futurebus+ Read Access (thru LIFE)
with Forced Bus Relinquish and Retry ($\overline{TA}$, $\overline{TEA}$, $\overline{BG}$)
until the Read Data is Available

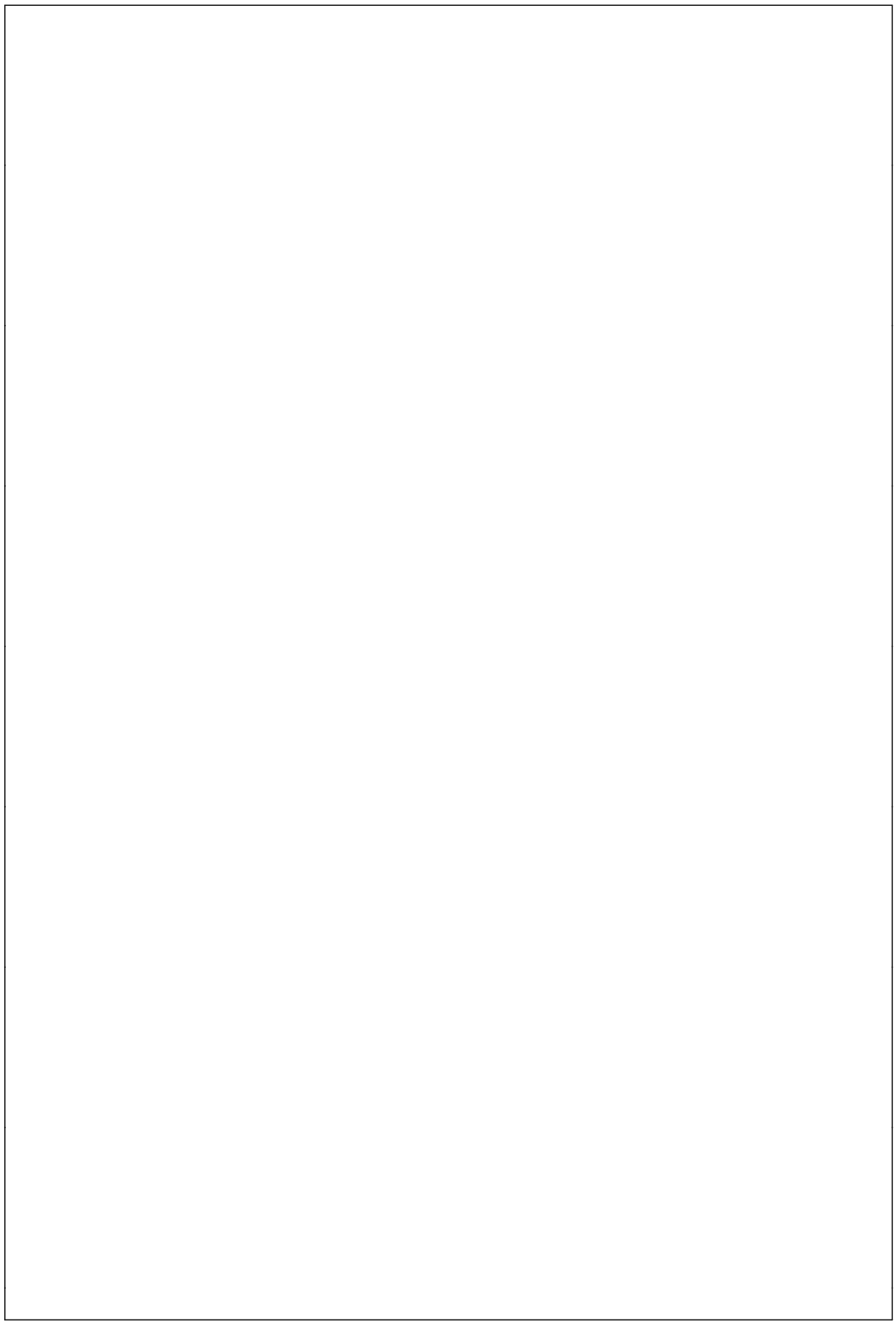


TL/F/11484–18

68040 Futurebus+ Read Access, Data Available



TL/F/11484–19



National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: 1(800) 272-9959
TWX: (910) 339-9240

National Semiconductor GmbH
Livry-Gargan-Str. 10
D-82256 Fürstenfeldbruck
Germany
Tel: (81-41) 35-0
Telex: 527649
Fax: (81-41) 35-1

National Semiconductor Japan Ltd.
Sumitomo Chemical
Engineering Center
Bldg. 7F
1-7-1, Nakase, Mihama-Ku
Chiba-City,
Ciba Prefecture 261
Tel: (043) 299-2300
Fax: (043) 299-2500

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductores Do Brazil Ltda.
Rue Deputado Lacorda Franco
120-3A
Sao Paulo-SP
Brazil 05418-000
Tel: (55-11) 212-5066
Telex: 391-1131931 NSBR BR
Fax: (55-11) 212-1181

National Semiconductor (Australia) Pty, Ltd.
Building 16
Business Park Drive
Monash Business Park
Nottinghill, Melbourne
Victoria 3168 Australia
Tel: (3) 558-9999
Fax: (3) 558-9998