Dual Port Interface for the DP8417/18/19/28/29 DRAM Controller

INTRODUCTION

This application note describes a general purpose dual port interface to the DP8417/18/19/28/29 DRAM controller. A PAL® (Programmable Array Logic) device is used to implement this interface. The PAL contains the logic necessary to arbitrate between the three ports (Refresh, Port A, and Port B), provide WAIT states to Port A or B when necessary, and an output to multiplex the Port A or B addresses to the DRAM controller.

FEATURES

■ Provides a versatile dual port interface to the DP8417/18/19/28/29 DRAM controller

■ Provides arbitration circuitry between DRAM refresh cycles, Port A accesses, and Port B accesses

■ Allows for Port A and Port B to be synchronous or asynchronous to the input system clock

■ Guarantees a minimum of one and one half system clock periods of RAS precharge time between grants to any two ports

■ Provides WAIT state logic to both PORT A and Port B to handle contention problems between ports

■ Differentiates between READ and WRITE accesses for Port A allowing Port A WRITE accesses to begin later than READ accesses

DESCRIPTION

This hardware arbitrates access to the dynamic RAM controlled by the DP8419 (or any of the related family members: DP8417/18/19/28/29) to either:

1) A Refresh cycle, "$\overline{\text{GRNTRF}}$"

2) Port A, "$\overline{\text{GRNTA}}$"

3) Port B, "$\overline{\text{GRNTB}}$"

Refresh always has the highest priority and will always occur immediately upon a refresh request ($\overline{\text{RFRQ}}$) given that an access by Port A or B is not currently in progress. Port A has a higher priority than Port B though the scheme used attempts to give both ports a more equal priority. The arbiter does this by leaving Port A or Port B granted, after an access by that particular port, as long as no other ports are currently trying to access the DRAM. This scheme is used because data tends to be transferred in bursts from a particular port.

Once a port is granted, subsequent requests by that port immediately access the DRAM, until another port gains access to the DRAM (see *Figure 11* of the timing waveforms for Port A).

The term "$\overline{\text{WINA}}$" (write enable for Port A) is used to cause "$\overline{\text{RASIN}}$" to be generated later for a WRITE access than a READ access. This may be necessary to guarantee that valid data is written to the DRAM during WRITE accesses. If Port B is asynchronous this input is not needed because Port B requests are delayed through the external synchronization circuitry. If Port B is synchronous both ports should mux to the "$\overline{\text{WIN}}$" input, and use this input in generating the "$\overline{\text{RASIN}}$" output of the PAL.

This arbiter guarantees one and one half system clock periods of RAS precharge between accesses of different ports. It is up to the user to guarantee the precharge time between consecutive accesses from the same port. This arbiter assumes a minimum of one period high time between access requests from a particular port.

Hidden Refresh is not supported in any of the following dual port schemes for several reasons:

1) If "$\overline{\text{CS}}$", of the DP8419, is not permanently tied low the user must guarantee a "$\overline{\text{CS}}$-$\overline{\text{RASIN}}$" minimum time of 34 ns for the DP8419. This could slow down the access time of several of the dual port schemes presented.

2) In order to do hidden refresh a port must be granted during a non-$\overline{\text{CS}}$ access cycle. When the port is granted during a non-$\overline{\text{CS}}$ access cycle the other port may be requesting the dual ported memory also and have to wait for it. A possible problem is that the non-$\overline{\text{CS}}$ access may not even be causing a hidden refresh at that time so in essence the other port is being slowed down for no reason (i.e. a hidden or forced refresh may have already been done during that period of the refresh clock).

If either Port A or B tries to access the DRAM during a refresh WAIT states will automatically be inserted into that port's access cycle. Also if one of the ports tries to access the DRAM while the other port is, WAIT states will automatically be inserted into the appropriate port's access cycle. The user may want to change the "$\overline{\text{WAIT}}$" state equations depending upon the processor or bus being interfaced to.

The DUAL PORT ARBITER gives access to the refresh cycle via the M2 ($\overline{\text{RFSH}}$) pin of the DP8419. The $\overline{\text{GRNTB}}$ output of the DUAL PORT CONTROLLER acts as a multiplexor signal to enable either PORT A or PORT B. Once enabled the Port selected will enable its addresses, write enable, $\overline{\text{LOCK}}$ control signal, and data to the DP8419 and its controlled memory. The user must be careful to assure that a particular port will not be locked ("$\overline{\text{LOCK}}$" low while "$\overline{\text{GRNTA}}$ or B" is low) for more than 15.6 μs (RFCK period) or the system may miss a refresh.

The Dual Port scheme presented assumes that all "PORT REQUEST" inputs are synchronous to the system clock input to the PAL (i.e. "PORT REQUESTs" occur following a rising edge of the system clock). If a specific "PORT RE-QUEST" is asynchronous to the system clock it has to be synchronized to the system clock by running it through two flip-flops (see "AREQB" and "ARFRQ" in the system block diagram). The two "$\overline{\text{RFRQ}}$" synchronizing flip-flops are needed for the PAL refresh logic to work correctly.

The Dual Port scheme presented does not assume the use of any specific processor. Therefore, the user may require some external logic to interface the Dual Port PAL to a specific microprocessor or bus.

Figures 1–5 show several suggestions for circuits used to generate "$\overline{\text{REQA}}$" for different CPU's. The PAL equations were designed assuming a National Semiconductor Series 32000® CPU on Port A. In the "$\overline{\text{RASIN}}$" equations for Port A WRITE cycles were started one half period later than READ

cycles and both READ and WRITE accesses were ended one half period after "$\overline{REQA}$" went high (this is to make up for WRITE accesses starting one half period after "$\overline{REQA}$"). The user may wish to modify these equations (and possibly the "$\overline{WAITA}$" equations) depending upon the specific CPU being used.

EXAMPLE: DETERMINING THE REQUIRED MEMORY SPEED ("t_{RAC}" AND "t_{CAC}") FOR A SERIES 32000 TO RUN AT 10 MHz WITHOUT WAIT STATES

Assume the Series 32000 is synchronously interfaced to Port A.

#1) $\overline{RASIN}$ low = T1 + 6 ns (PHI1 to CTTL Rising edge maximum) + 12 ns ("B" PAL clocked output) + 15 ns ("B" PAL combinational output) = 100 + 6 + 12 + 15 = 133 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 70 ns (DP8419-70) − 3 ns (72 DRAMs instead of 88 DRAMs spec'd in data sheet) = 67 ns maximum

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" clock cycle = 15 ns maximum

"t_{RAC}" = T1 + T2 + T3 − #1 − #2 − #4 − #5
       = 100 + 100 + 100 − 133 ns − 20 − 7 − 15 = 125 ns

"t_{CAC}" = T1 + T2 + T3 − #1 − #3 − #4 − #5
       = 100 + 100 + 100 − 133 ns − 67 − 7 − 15 = 78 ns

Therefore the DRAM chosen should have a "t_{RAC}" less than or equal to 125 ns and a "t_{CAC}" less than or equal to 78 ns. Standard 120 ns DRAMs meet this criteria.

The following is an example of how to interpret the PAL equations correctly. These equations are presented in the format specified by the National Semiconductor PLAN format. CAUTION, this format differs from the much used PALASM format.

EXAMPLE: $\overline{GRNTRF}$:= $\overline{RFRQ}$*GRNTA*GRNTB

This reads, the active low flip-flop output "$\overline{GRNTRF}$" is low following the rising edge of the input clock given that, the active low input "$\overline{RFRQ}$" is low AND the active low output "$\overline{GRNTA}$" is high AND the active low output "$\overline{GRNTB}$" is high a setup time before the input clock transitions high. (Notice that $\overline{RFRQ}$ is interpreted as being low.)

POSSIBLE MODIFICATIONS TO THIS APPLICATION

In this application "$\overline{REQB}$" is synchronized to the falling edge of the system clock input of the PAL. Generating "$\overline{REQB}$" from the falling clock edge allows minimum delay from the asynchronous request to the synchronized request producing "$\overline{GRNTB}$" and or "$\overline{RASIN}$". Producing "$\overline{REQB}$" in this way also delays "$\overline{RASIN}$" during a port B access because of the effect of the "$\overline{GTOA}$" term. In order to calculate the t_{RAC} and t_{CAC} of the DRAM (see Series 32000 example above) the delay to "$\overline{RASIN}$" low would be: "$\overline{AREQB}$" low (asynchronous request B) + SYNCHRONIZATION delay (2 flip-flops) + 3 input NAND gate delay of "$\overline{GTOA}$" + PAL delay for "$\overline{RASIN}$".

If "$\overline{REQB}$" is synchronized to the rising edge of the system clock there is a potential danger of getting glitches on the "$\overline{RASIN}$" output of the PAL as a result of the "$\overline{GTOA,B}$" terms. The glitches are possible under the condition of both "$\overline{REQA}$" and "$\overline{REQB}$" going low during a single clock period. For example, if Port B is currently granted ("$\overline{GRNTB}$" low) and "$\overline{REQA}$" goes low more than one inverter gate delay before "$\overline{REQB}$" goes low the "$\overline{GTOA}$" term will initially be high, then go low, then back high. This could cause a small glitch at the beginning of "$\overline{RASIN}$". This glitch can be avoided by guaranteeing that either the requests are separated by at least a three input NAND gate delay (as is the case in this application note) or that when two requests happen within one clock period they happen within one inverter gate delay of each other. The circuits shown below, in *Figure 1*, could be used to guarantee that when two requests happen within one clock they occur within one gate delay of each other.
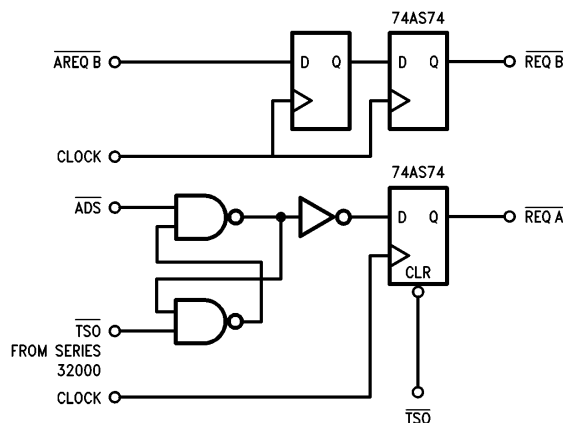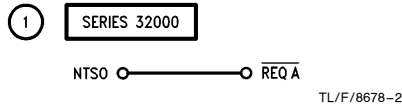

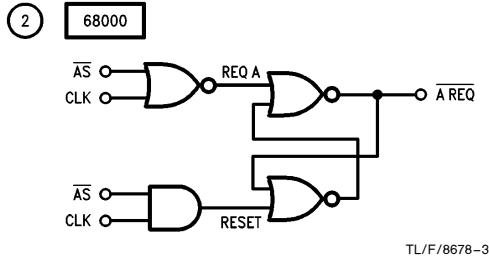FIGURE 1. Alternative Request Generating Circuits

TL/F/8678−1

2

IDEAS ON GENERATING "$\overline{\text{REQA}}$" FOR SEVERAL DIFFERENT MICROPROCESSORS.

*$\overline{\text{REQA}}$, $\overline{\text{REQB}}$, $\overline{\text{RFRQ}}$ should have a minimum setup time of approximately 20 ns before the rising edge of the system clock.

FIGURE 2. Series 32000 "$\overline{\text{REQA}}$"

Minimum of 2 periods $\overline{\text{RAS}}$ precharge between successive accesses.

FIGURE 3. 68000 "$\overline{\text{REQA}}$"

Minimum of 1½ periods of $\overline{\text{RAS}}$ precharge.
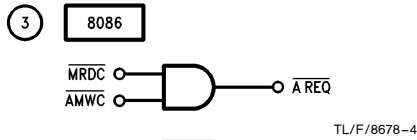
FIGURE 4. 8086 "$\overline{\text{REQA}}$" Method #1

Minimum of 2 periods of $\overline{\text{RAS}}$ precharge.
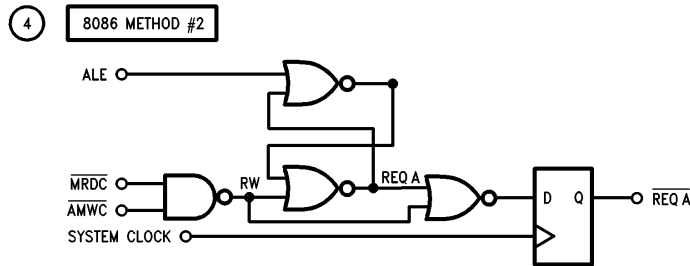
FIGURE 5. 8086 "$\overline{\text{REQA}}$" Method #2

(For faster speed, minimum of 1 period of $\overline{\text{RAS}}$ precharge.)

DUAL PORT PAL #1 INPUTS

1) "CLOCK" System clock.

2) "$\overline{\text{REQA}}$" A synchronous access request from Port A.

3) "$\overline{\text{WINA}}$" WRITE ENABLE from Port A. This input is used to delay "$\overline{\text{RASIN}}$" during WRITE accesses.

4) "$\overline{\text{REQB}}$" A synchronous chip selected access request form Port B. "$\overline{\text{AREQB}}$" is run through two flip-flops to get "$\overline{\text{REQB}}$". Chip Select for Port B is assumed to be included within this input.

5) "$\overline{\text{RFRQ}}$" A synchronous refresh request.

6) "$\overline{\text{LOCK}}$" The "$\overline{\text{LOCK}}$" input is an active low signal that is driven by either Port A or Port B. This input, when low, causes the arbiter to keep the currently granted Port granted until the "$\overline{\text{LOCK}}$" input goes high. This input is useful in implementing atomic operations such as semaphores that are useful in multiuser/multitasking operating systems.

7) "$\overline{\text{GTOA}}$" This input is generated externally using the three signals $\overline{\text{REQA}}$, $\overline{\text{REQB}}$, and $\overline{\text{LOCK}}$ with some discrete logic. This input indicates that the arbiter will switch to Port A, given that Port B is currently granted. This input is needed to guarantee that when the arbiter switches control of the DRAM from Port B to Port A that $\overline{\text{GRNTB}}$ goes invalid before $\overline{\text{REQB}}$ is able to start another access (see the $\overline{\text{RASIN}}$ output term "PORTB RASIN" in PAL equations).

8) "$\overline{\text{GTOB}}$" This input is generated externally using the three signals $\overline{\text{REQA}}$, $\overline{\text{REQB}}$, and $\overline{\text{LOCK}}$ with some discrete logic. This input indicates that the arbiter will switch to Port B given that Port A is currently granted. This input is needed to guarantee that when the arbiter switches control of the DRAM from Port A to Port B that $\overline{\text{GRNTA}}$ goes invalid before $\overline{\text{REQA}}$ is able to start another access (see the $\overline{\text{RASIN}}$ output term "PORTA RASIN" in PAL equations).

9) "CLK" This is the system clock input that may be used in the PAL equations (i.e. "$\overline{\text{WAIT}}$").

10) "$\overline{\text{CSA}}$" This input is the chip select input for Port A. It is used, along with "$\overline{\text{REQA}}$", to request and cause an access to the DRAM.

DUAL PORT PAL #1, OUTPUTS

NOTE: All outputs are active low.

1) "$\overline{\text{RASIN}}$" This is the $\overline{\text{RASIN}}$ input to the DP8419 for Port A, Port B, and refresh.

2) "$\overline{\text{GRNTA}}$" This output is the grant output for Port A.

3) "$\overline{\text{GRNTB}}$" This output functions as the grant output for both Port A (high) and Port B (low).

4) "$\overline{\text{GRNTRF}}$" Goes to DP8419 M2 ($\overline{\text{RFSH}}$) input. This causes an automatic forced refresh cycle.

5) "$\overline{\text{GRNT1D}}$" Goes low one period after "$\overline{\text{GRNTA}}$", "$\overline{\text{GRNTB}}$", or "$\overline{\text{GRNTRF}}$" go low. This output is used to guarantee that one period is allowed after arbitration before a "$\overline{\text{RASIN}}$" is generated during a port access. This allows the particular port's address, write enable signal, and lock input to become valid before an access is started. This output also allows the PAL to determine when a particular port has been granted for several system clock periods. This information allows the arbiter to immediately generate "$\overline{\text{RASIN}}$" for any subsequent memory accesses since the address is already muxed to the DRAM controller (see *Figure 11* for the timing waveforms for Port A).

6) "$\overline{\text{WAITA}}$" This output functions as a WAIT input for Port A.

7) "$\overline{\text{GTORFSH}}$" This input is generated internally and indicates that the arbiter will give access control over to the refresh Port at the next rising clock edge.

8) "$\overline{\text{XACKB}}$" This output is generated external to the PAL and functions as a transfer acknowledge for Port B.

```
DUAL PORT PAL #1
PAL16R4B


CLOCK  /REQA  /WINA  /REQB  /RFRQ  /LOCK  /GTOA  /GTOB  CLK   GND
/OE  /CSA  /WAITA  /GRNT1D  /GRNTRF  /GRNTB  /GRNTA  /RASIN  /GTORFSH   VCC


/GRNTA  := /CSA*/REQA*GRNTRF*RFRQ*GRNTB      ;Start GRNTA
         + /LOCK*/GRNTA                      ;Continue GRNTA
         + /CSA*/REQA*RFRQ*/GRNTRF*GRNT1D    ;RFSH_TO_PORTA
         + /CSA*/GTOA*/*GRNTB*RFRQ*RASIN     ;PORTB_TO_PORTA
         + /CSA*/REQA*/GRNTA                 ;Hold GRNTA
         + /GRNTA*REQB*RFRQ                  ;Hold GRNTA


/GRNTB  := REQA*GRNTA*RFRQ*GRNTRF*/REQB      ;Start GRNTB
         + /LOCK*/GRNTB                      ;Continue GRNTB
         + /GTOB*/GRNTA*RFRQ*RASIN           ;PORTA_TO_PORTB
         + REQA*RFRQ*/GRNTRF*/REQB*GRNT1D    ;RFSH_TO_PORTB
         + /REQB*/GRNTB                      ;Hold GRNTB
         + /GRNTB*REQA*RFRQ                  ;Hold GRNTB
         + /GRNTB*CSA*RFRQ                   ;Hold GRNTB


/GRNTRF := GRNTA*GRNTB*/RFRQ                 ;Start GRNTRF
         + /GRNTRF*/RFRQ                     ;Continue GRNTRF
         + REQA*/GRNTA*LOCK*/RFRQ            ;PORTA_TO_RFSH
         + REQB*/GRNTB*LOCK*/RFRQ            ;PORTB_TO_RFSH
         + /GRNTRF*/GRNT1D                   ;Hold GRNTRF


/GRNT1D := /GRNTA*GTOB*GTORFSH              ;GRNT1D for PORTA
         + /GRNTB*GTOA*GTORFSH              ;GRNT1D for PORTB
         + /GRNTRF*/RFRQ                    ;GRNT1D for RFSH


IF (VCC) /GTORFSH =
         REQA*/GRNTA*LOCK*/RFRQ             ;PORTA_TO_RFSH
        +REQB*/GRNTB*LOCK*/RFRQ             ;PORTB_TO_RFSH
```

```
DUAL PORT PAL #1,
PAL16R4B (Continued)
IF (VCC) /RASIN =
/CSA*/REQA*/GRNTA*/GRNT1D*GTOB*GTORFSH*WINA          ;PORTA READ RASIN
+/CSA*/REQA*/GRNTA*/GRNT1D*GTOB*GTORFSH*/WINA*/CLK   ;PORTA WRITE RASIN
+/CSA*/REQA*/GRNTA*/GRNT1D*GTOB*GTORFSH*/WINA*/RASIN ;PORTA WRITE RASIN
+/GRNTA*/GRNT1D*/RASIN*CLK                           ;Hold PORTA RASIN
+/REQB*/GRNTB*/GRNT1D*GTOA*GTORFSH                   ;PORTB RASIN
+/GRNTRF*/GRNT1D*GTORFSH                             ;RFSH RASIN

IF (VCC) /WAITA = /CSA*/REQA*GRNTA             ;Start WAITA
         +/CSA*/REQA*/WAITA*CLK                ;WAIT until one half period after GRNTA
```
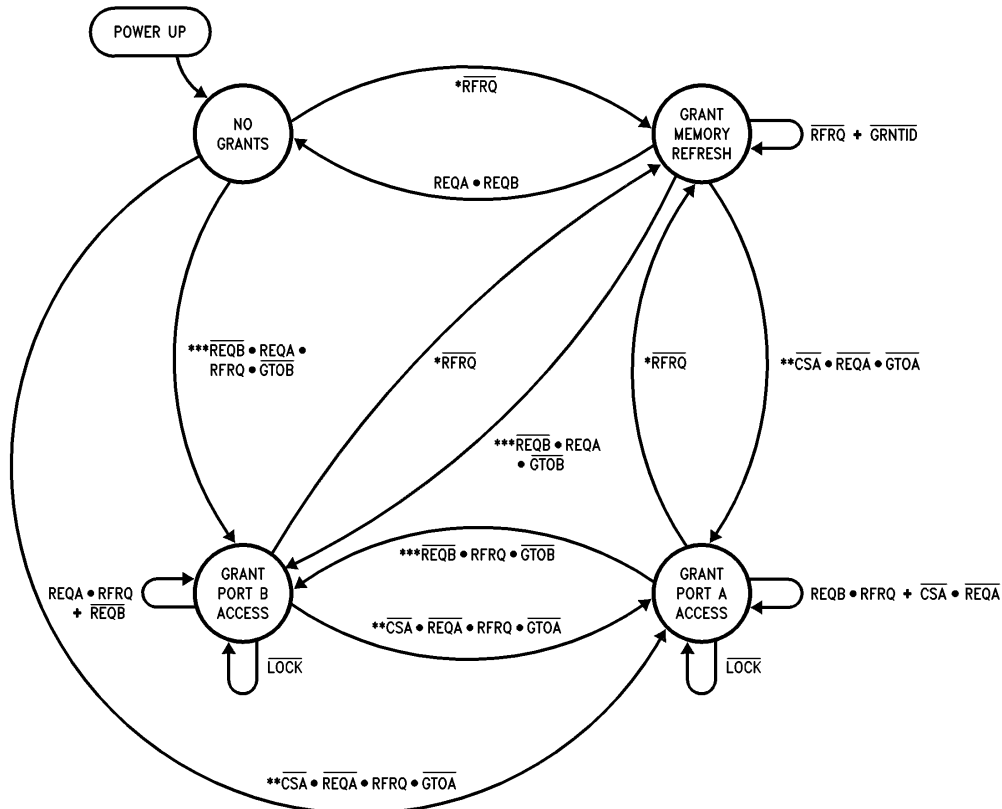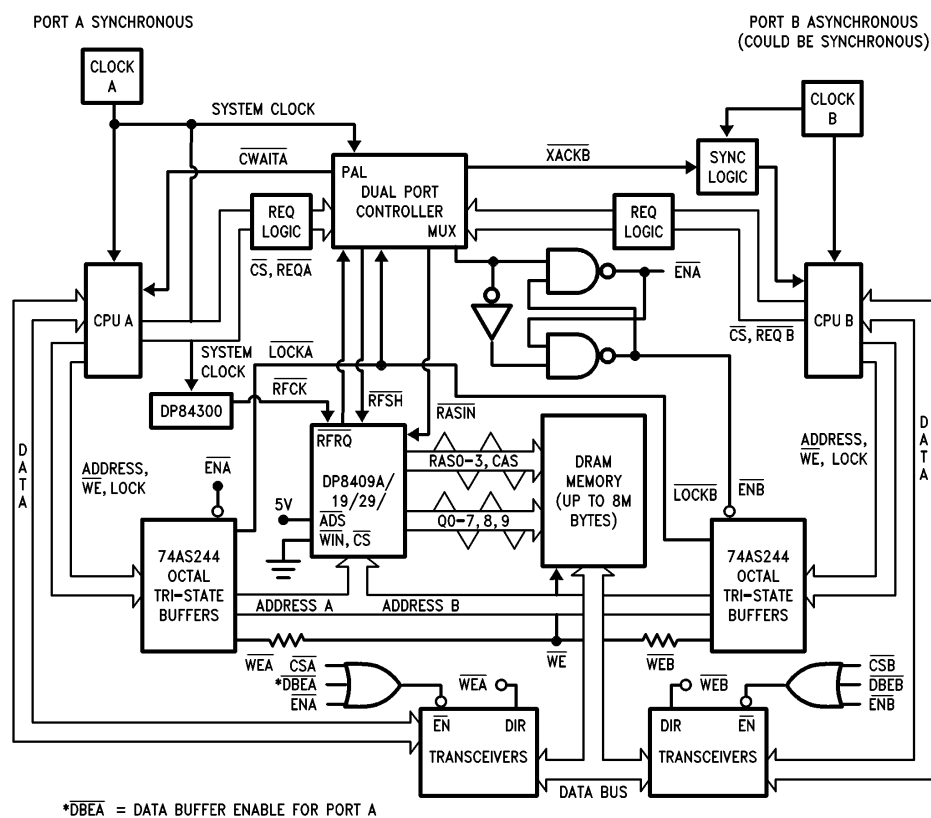

FIGURE 6. Dual Port State Diagram

TL/F/8678–6

*Refresh has the highest priority
**Port A has the second highest priority
***Port B has the lowest priority

5

FIGURE 7. Dual Port Interface
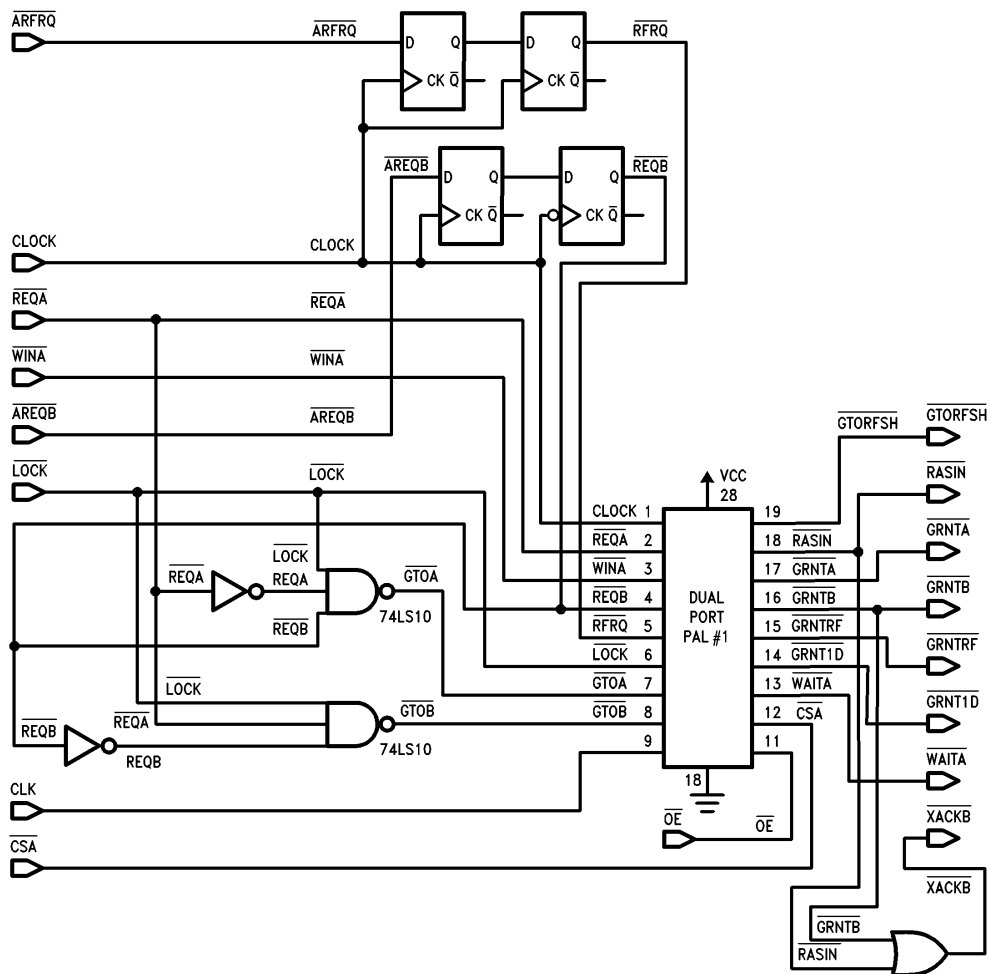
TL/F/8678-7

FIGURE 8. Dual Port PAL Controller Diagram
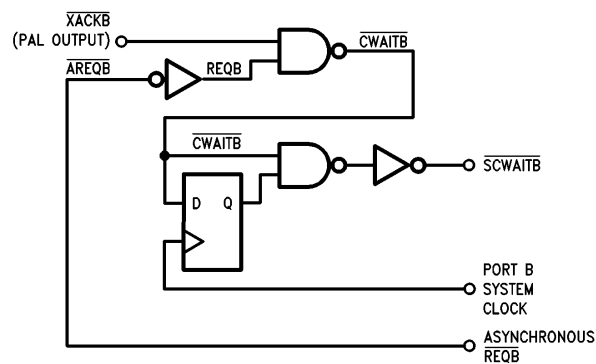
TL/F/8678–8

TL/F/8678–9

FIGURE 9. Asynchronous Port B transfer acknowledge ("$\overline{XACKB}$") synchronizes circuit to produce "CWAITB" synchronous with the Port B clock "CTTL" ("$\overline{SCWAITB}$")
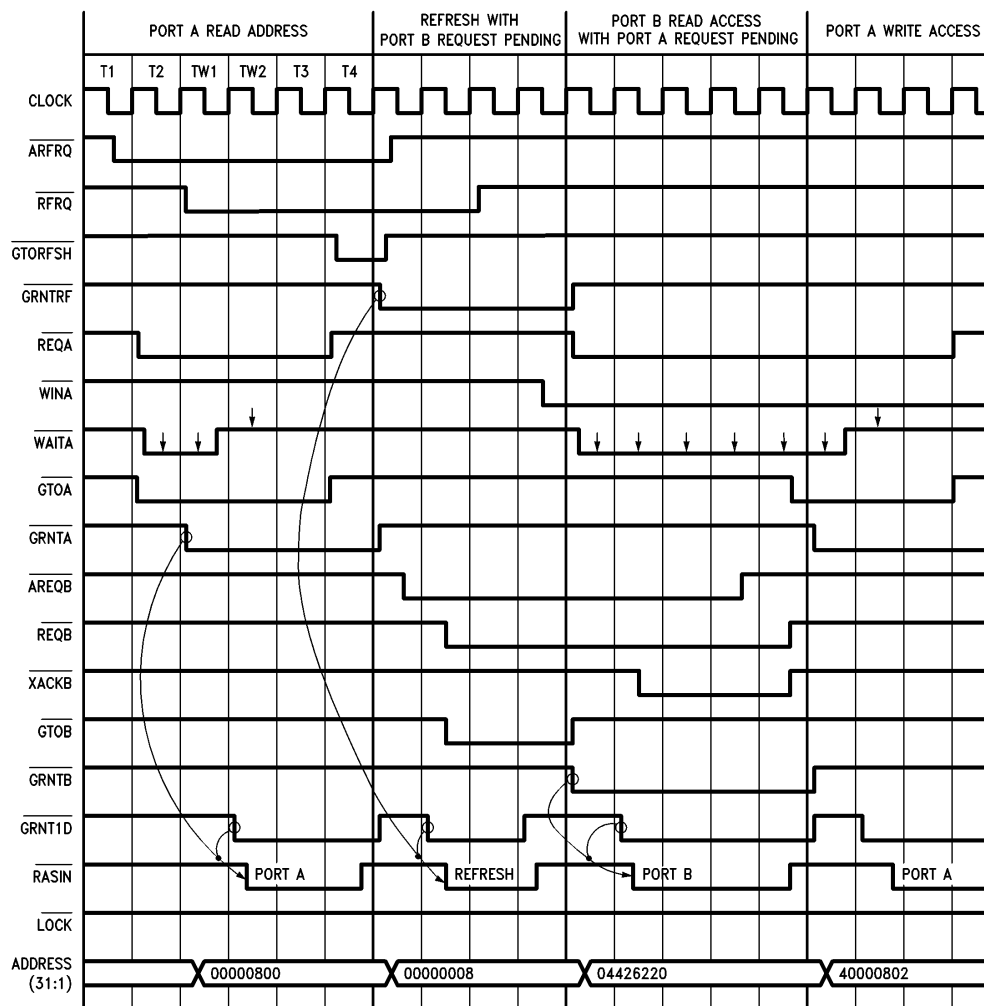
FIGURE 10. Dual Port Timing

TL/F/8678–10

8

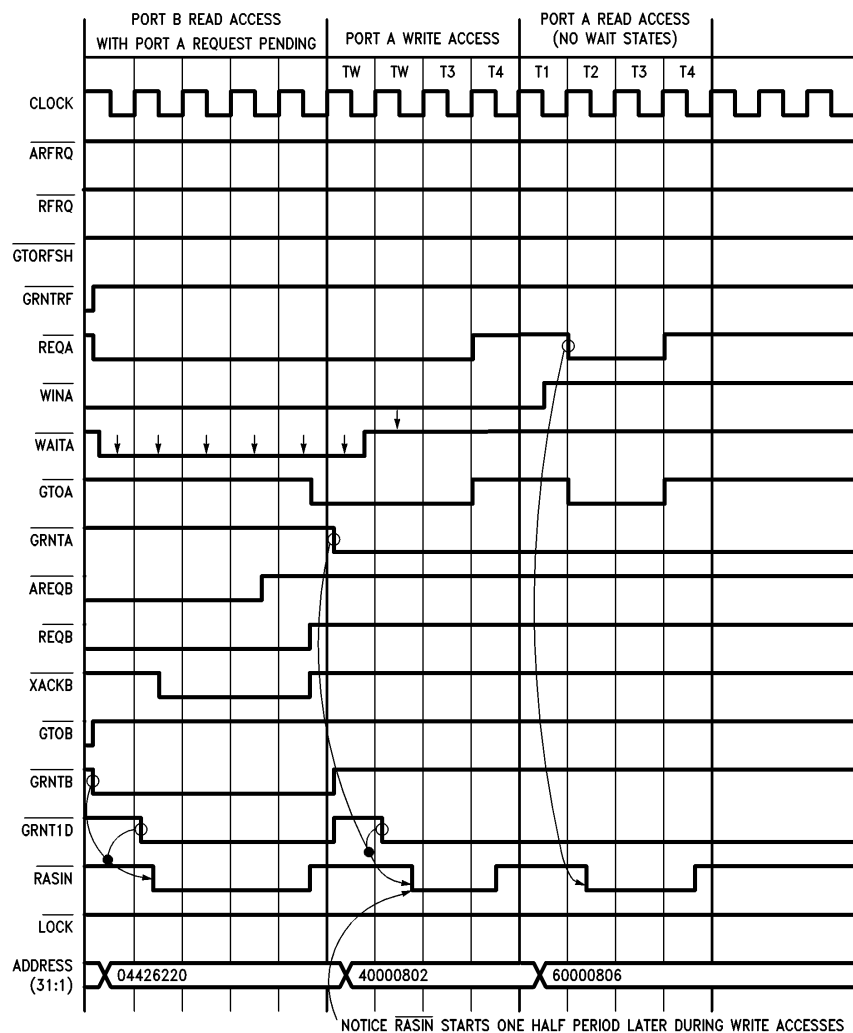**FIGURE 11. Dual Port Timing**

TL/F/8678–11

Lit. # 100436

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.