National Semiconductor

PRELIMINARY

DP84522 Dynamic RAM Controller Interface Circuit for the 68020 CPU

General Description

This is a Programmable Array Logic (PAL®) device designed to allow an easy interface between the 68020 microprocessor and the National Semiconductor DP8417, DP8418, DP8419, DP8428 or DP8429 dynamic memory controllers.

This PAL supplies all the control signals needed to perform memory read, write, and refresh operations up to a frequency of 16.7 MHz.

Features

- Provides a 3 or 4 chip solution for the 68020/DP8418 (or DP8428) dynamic RAM interface (1 or 2 PALs, DP8418, and clock divider)
- Works with all speed versions of the 68020 up to 16.7 MHz
- Allows operation of 68020 at 12.5 MHz with 1 WAIT state with standard 120 ns 256k DRAMs
- Controls DP8418/28 mode 5 accesses and mode 1 or 0 forced refreshes automatically
- Allows memory interleaving if desired
- CPU WAIT states are automatically inserted during contention between memory interleaving/DRAM accesses/ DRAM refreshes
- Uses standard National Semiconductor PALs (i.e., DMPAL16R4A; the user may want to use faster versions of these PALs at higher CPU operating frequencies)
- The PAL programming equations are provided with comments for easy user modification to his specific requirements

Functional Description

The following description applies only to the DP8418 or DP8428 since "RFI/O" going low initiates a mode 0 externally controlled forced refresh. This forced refresh resets the internal refresh request logic on the DP8418 or DP8428, but will not reset the internal logic on the DP8409A.

A memory cycle starts when chip select ($\overline{CS}$) and the address strobe ($\overline{AS}$) become true. $\overline{RASIN}$ is supplied from the PALs to the DP8418 DRAM controller, which then supplies $\overline{RAS}$ to the selected $\overline{RAS}$ bank. After the necessary row address hold time, the DP8418 switches the address outputs to the column address. The DP8418 then supplies the required $\overline{CAS}$ signal to the DRAM.

The first PAL (PAL #1) supports byte operations by producing four WRITE enables, one for each possible byte of the 32 bit word (upper, upper middle, lower middle, and lower write enable). These WRITE enables are produced externally from the 68020 "DATA STROBE" and "READ/WRITE" outputs. Since it is possible that all WRITE cycles may be LATE WRITE cycles ("WRITE ENABLE" occurring after "COLUMN ADDRESS STROBE") memory buffers should be used instead of transceivers to separate the data in from the data out of the DRAMs.

The second PAL (PAL #2) supports byte operations by producing four COLUMN ADDRESS STROBES, one for each of the possible bytes of the 32-bit word. This PAL terminates the DP8418 "$\overline{RASIN}$" input early but holds the DRAM data valid by latching the byte "$\overline{CAS}$'s" externally. This method of supporting byte writes allows transceivers to be used, or to directly connect the DRAM data in and data out pins to the 68020 data bus I/O pins.

Hidden REFRESH cycles are not allowed in this set of PALs because of the need for adequate $\overline{RAS}$ precharge times in all circumstances and the desire not to be inserting WAIT states into access cycles of other system elements.

These PALs perform externally controlled forced refreshes automatically (mode 0). A refresh cycle occurs when the DP8418 input RFCK transitions low and the RFIO signal goes low requesting a refresh cycle. The PAL responds by pulling RFSH low (M2 and M0) if there are no current DRAM accesses in progress. If a DRAM access is in progress the PAL waits until the current access is completed before performing the forced refresh cycle. If an access is requested during the forced refresh cycle WAIT states are automatically inserted into the access cycle until the refresh cycle is completed and adequate $\overline{RAS}$ precharge has been completed. The pending DRAM access cycle is then performed. The input signal "$\overline{NOWAIT}$" allows one to vary the amount of time required to do a refresh (see the description of the "$\overline{NOWAIT}$" input in the pin description section). In one of the timing diagrams the "$\overline{RFSH}$" output was tied to the "$\overline{NOWAIT}$" input to decrease the length of the refresh cycle but still insert one wait state in normal DRAM access cycles (see *Figure 5*).

The first PAL (PAL #1) supports memory interleaving to guarantee adequate $\overline{RAS}$ precharge time during two consecutive accesses to the same DRAM bank. This is performed by looking at the lower address bit or bits, A2 and/or A3. If the processor is sequentially accessing the DRAM each $\overline{RAS}$ output will have plenty of precharge time. But if the system tries to access the same bank twice in a row the access will be delayed until adequate $\overline{RAS}$ precharge time has been met. During this time WAIT states will automatically be inserted into the pending access cycle.

The second PAL (PAL #2) guarantees adequate $\overline{RAS}$ precharge time (one and one half system clock periods) by ending the DP8418 "$\overline{RASIN}$" input early. The DRAM data is held valid by externally latching the DRAM "$\overline{CAS}$" input as explained earlier. This has the additional benefit of sim-

Functional Description (Continued)

plifying the memory interface of the 68020 by eliminating the external components needed for interleaving, though as one approaches 16.7 MHz the interleaving circuitry may again become necessary to guarantee adequate "$\overline{RAS}$" precharge time.

For PAL #1 an external "D type" flip-flop or another PAL could be used for the support of memory interleaving. If one is not using memory interleaving (10 MHz or below) the "PREVO" input can be used for some other function and the equations of "$\overline{RASIN}$" that employ "PREVO" can be adjusted.

The PAL equations for this interface are written in the National Semiconductor PLAN™ format, which differs from the standard PALASM™ format.

EXAMPLE: PLAN FORMAT

```
/RASIN := RFSH*/2D*/AS
```

This translates as, "$\overline{RASIN}$" is low after the rising edge of the input clock given that "$\overline{RFSH}$" was high and "$\overline{2D}$" was low and "$\overline{AS}$" was low a setup time before the clock transitions high (here $\overline{RASIN}$, $\overline{2D}$, and $\overline{RFSH}$ are outputs of the PAL and $\overline{AS}$ is an input).

EXAMPLE: PALASM FORMAT

```
RASIN := /RFSH*2D*AS
```

The above expression means the same as the PLAN format expression except it is written in PALASM format. In other words; "$\overline{RASIN}$" will go low after the rising edge of the clock given that "$\overline{RFSH}$" was high, "$\overline{2D}$" was low, and "$\overline{AS}$" was low a setup time before the clock transitions high (here $\overline{RASIN}$, $\overline{2D}$, and $\overline{RFSH}$ are outputs of the PAL and $\overline{AS}$ is an input).

Depending on the specific type of PALs and logic used the user can calculate the speed requirements for the DRAM at the specified processor frequency as follows:

CALCULATION OF DRAM "t_{RAC}" $\overline{RAS}$ ACCESS TIME AND "t_{CAC}" $\overline{CAS}$ ACCESS TIME REQUIRED FOR A 12.5 MHz 68020, 1 WAIT STATE, MICROPROCESSOR SYSTEM

#1) $\overline{RASIN}$ generation time = "S0" + "S1" + 1 combinational output delay of the PAL generating the "$\overline{RASIN}$" output (assume DMPAL16R4B) = 80 ns + 15 ns = 95 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ out delay of the DP8418 = 20 ns maximum (used to determine "t_{RAC}")

#3) $\overline{RASIN}$ to $\overline{CAS}$ out delay of the DP8418 DRAM contoller driving a load of 2 banks of 256k DRAMs, each bank containing 36 (32 DRAMs plus byte parity) = 72 DRAMs

Since this is under the specified load in the data sheet (approximately 88 DRAMs) approximately 3 ns can be subtracted from the data sheet number, giving 80 ns − 3 ns = 77 ns maximum (used to determine "t_{CAC}")

#4) 74AS244 buffer delay = 7 ns maximum

#5) Data setup time required from the falling edge of "S4" clock = 10 ns maximum

A normal 12.5 MHz 68020 access cycle (with 1 WAIT state inserted) contains 4 clock periods of 80 ns per period.

The required DRAM "t_{RAC}" (row access time) can be calculated from

S0 + S1 + S2 + S3 + SW1 + SW2 + S4
(minimum 1/2 period) − #1 − #2 − #4 − #5
= 40 + 40 + 40 + 40 + 40 + 40 + 35
− 95 − 20 − 7 − 10 = 143 ns

The required DRAM "t_{CAC}" (column access time) can be calculated from

S0 + S1 + S2 + S3 + SW1 + SW2 + S4
(minimum 1/2 period) − #1 − #3 − #4 − #5
= 40 + 40 + 40 + 40 + 40 + 40 + 35
− 95 − 77 − 7 − 10 = 86 ns

The DRAMs selected for this system must satisfy both the "t_{RAC} and t_{CAC}" requirements. Therefore the DRAMs must have a "t_{RAC}" (row access time) less then or equal to 143 ns and a "t_{CAC}" (column access time) less than or equal to 86 ns to be used in this system, under worst case conditions, for a 1 WAIT state 12.5 MHz 68020 system. Common 120 ns 64k or 256k DRAMs meet this specification. If one is using PAL #2, producing external "CAS's" and not using any external transceivers he could possibly use 150 ns DRAMs in the above example.

If one is using PAL #2, the calculated "t_{RAC}" and "t_{CAC}" may differ from the actual system values, depending upon the external circuitry used to produce the byte "$\overline{CAS}$'s". The DP8418 "$\overline{RASIN}$–$\overline{CAS}$" low will be approximately 10–15 ns less than the value given in the data sheet because of the small loading on the DP8418 "$\overline{CAS}$" output. The external circuitry needed to produce the byte "$\overline{CAS}$'s" should be loaded such that the column address (from DP8418) is valid when "$\overline{CAS}$" goes low. For this reason "$\overline{RASIN}$-byte $\overline{CAS}$" may be longer than the value used in the "t_{RAC}, t_{CAC}" calculations, and therefore may give a smaller "t_{RAC}, t_{CAC}" than was calculated.

68020 PAL Inputs and Outputs

(Pin number of the PAL on the left)

PAL #1 Inputs

1) "CK"	This is the system clock.
2) "$\overline{AS}$"	Address strobe from 68020.
3) "$\overline{RFRQ}$"	This is the refresh request from the DP8418.
4) "$\overline{CS}$"	This is the chip select (see system block diagram).
5) "R"	READ/$\overline{WRITE}$ output pin from the 68020.
6) "CLK"	The system clock.
7) "PREVO"	This output holds the previously accessed DRAM "$\overline{RAS}$" bank.
8) "B0"	This input is the address bit "A2" and is used to determine which "$\overline{RAS}$" bank the system is accessing.
9) "$\overline{NOWAIT}$"	This PAL always inserts one WAIT state into every 68020 access cycle. This input,

68020 PAL Inputs and Outputs (Continued)

if low, allows the DRAM to be accessed with no wait states inserted into the access cycle. This input also, if low during a refresh, shortens the length of the refresh cycle by one clock period. This causes the $\overline{RAS}$ pulse width (during a refresh) and the $\overline{RAS}$ precharge time (after a refresh) to be shorter.

11) "$\overline{OE}$" This input enables the PAL outputs.

PAL #1 Outputs

19) "$\overline{XDLY}$" This signal is used to guarantee one period of "$\overline{RFSH}$" high to "$\overline{RASIN}$" low time and to guarantee two periods of RAS precharge time in consecutive accesses to the same DRAM bank.

18) "$\overline{RASIN}$" This signal causes $\overline{RAS}$ (or $\overline{RAS}$s) to go low during a DRAM access or refresh.

17) "$\overline{RFSH}$" This signal initiates a DRAM Refresh.

16) "$\overline{1DLY}$" A delay that is used internally.

15) "$\overline{2DLY}$" A delay that is used internally.

14) "$\overline{RFREQ}$" Refresh request (from the DP8418) synchronized to the system clock.

13) "$\overline{RFREQCK}$" This input synchronizes "$\overline{RFREQ}$" to the falling edge of the input system clock "CLK" and is used in arbitrating between refreshes and accesses (see "$\overline{RASIN}$" equations).

12) "$\overline{DSACK}$" This output goes to the 68020 "$\overline{DSACK0}$, $\overline{1}$" data acknowledge input. This output allows WAIT states to be inserted into DRAM access cycles during access/refresh/RAS precharge contention.

Interface PAL #1 Boolean Equations

This PAL will work up to 16.7 MHz with the 68020. This PAL uses mode 0 (M0 = M1 = M2 = low) for doing externally controlled forced refreshes, guaranteeing more than 2.5 periods of RGCK RAS pulse width ("NOWAIT" = high). If "NOWAIT" is low the refresh is shortened by one clock period. This PAL will only work with the DP8417/18/19/28/29 since it uses mode 0 to reset the RFSH request (RFIO) signal.

```
DMPAL16R4A
CK   AS   RFRQ   CS   R   CLK   PREV0   B0   /NOWAIT   GND
/OE  /DSACK  /RFREQCK  /RFREQ  /2DLY  /1DLY  /RFSH  /RASIN  /XDLY  VCC

IF (VCC) /XDLY = RFSH*/2DLY*RASIN*PREV0*B0      ;Same bank interleave
     +RFSH*/2DLY*RASIN*/PREV0*/B0               ;Same bank interleave
     +/RFSH*1DLY*/2DLY*/RASIN*NOWAIT*/CLK       ;"/XDLY" low during RFSH
     +/RFSH*1DLY*2DLY*RASIN*RFREQ*/NOWAIT       ;"/XDLY" low during RFSH
     +/XDLY*/RFSH*RFREQ                         ;Hold "/XDLY" low
     +/XDLY*RASIN*/AS*CLK                       ;Hold "/XDLY" low

IF (VCC) /RASIN = /RFSH*RFREQ*/1DLY            ;RFSH "/RASIN"
  +/RASIN*/RFSH*/2DLY*XDLY                     ;Hold "/RASIN" low
  +RFSH*RFREQCK*/AS*/CS*PREV0*/B0*CLK          ;Start "/RASIN"
  +RFSH*RFREQCK*/AS*/CS*/PREV0*B0*CLK          ;Start "/RASIN"
  +RFSH*RFREQCK*/AS*/CS*2DLY*XDLY*CLK          ;After idle states
  +/RASIN*RFSH*/AS*/CS                         ;Hold "/RASIN" low
  +/RASIN*/CLK                                 ;Hold "/RASIN" low

IF (VCC) /RFREQCK = /RFREQ*/CLK                ;Start from falling clock
     +/RFREQCK*/RFREQ                          ;  edge

IF (/CS) /DSACK = /CS*RFSH*/RASIN*NOWAIT*/CLK  ;One WAIT state
  +/DSACK*/CS*RFSH*/RASIN*/AS                  ;Hold "/DSACK" low
  +/CS*RFSH*/AS*/NOWAIT*XDLY                   ;No WAIT state in access

/RFSH := /RFREQ*RASIN*/1DLY*/2DLY              ;Start RFSH
     +/RFREQ*RASIN*1DLY                        ;Start RFSH
     +/RFSH*/RFREQ                             ;Hold RFSH low
     +/RFSH*/RASIN                             ;Hold RFSH low
     +/RFSH*/1DLY                              ;Hold RFSH low

/1DLY := /RFSH*2DLY*/RFREQ                 ;Start "/1DLY" during RFSH
     +/RFSH*/1DLY*2DLY*XDLY                ;Continue "/1DLY" during RFSH
     +RFSH*/RASIN                          ;Start "/1DLY" during /RASIN

/2DLY := /RFSH*/1DLY                       ;Start "/2DLY" during RFSH
     +/RFSH*2DLY*/RFREQ*/NOWAIT            ;Shorten RFSH
     +RFSH*/RASIN*/1DLY                    ;Start "/2DLY" during /RASIN
     +RFSH*/RASIN*/NOWAIT                  ;Shorten access

/RFREQ := /RFRQ                            ;Synchronize to system clock
```

Interface PAL # 1 Between 68020 and DP8418/28

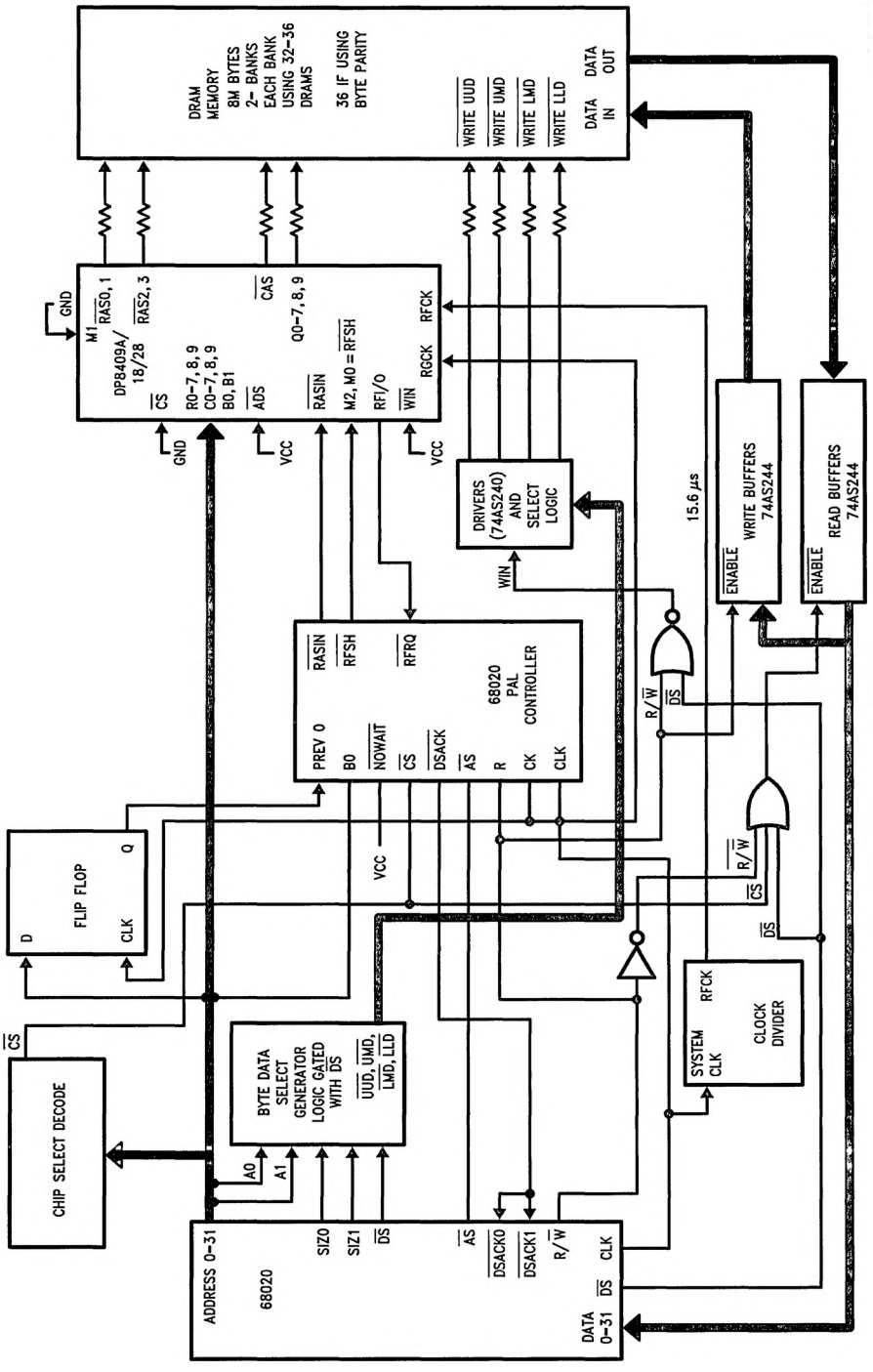
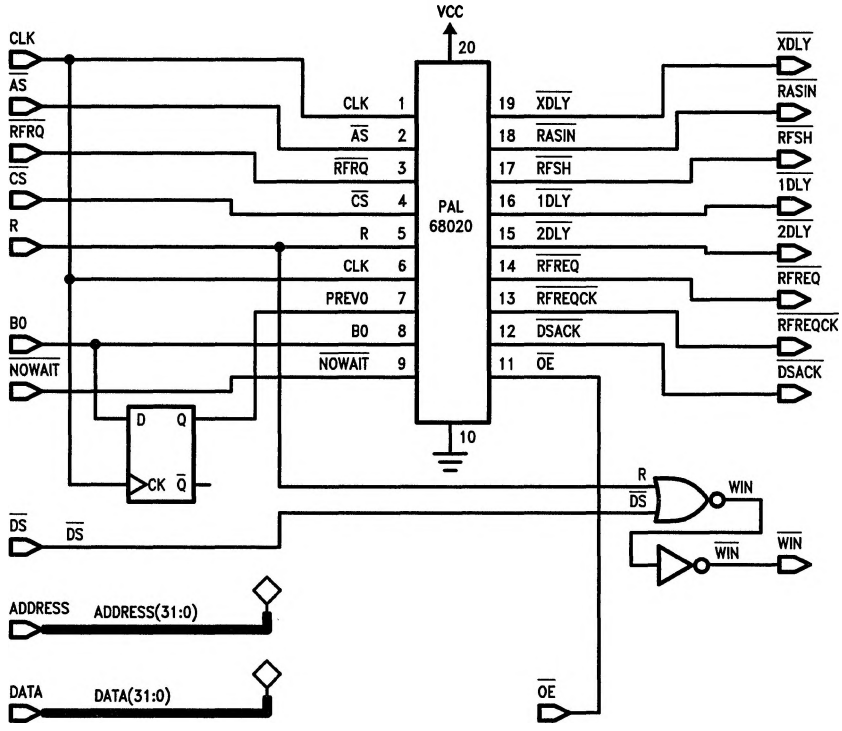


FIGURE 1. 68020 System Diagram for PAL # 1

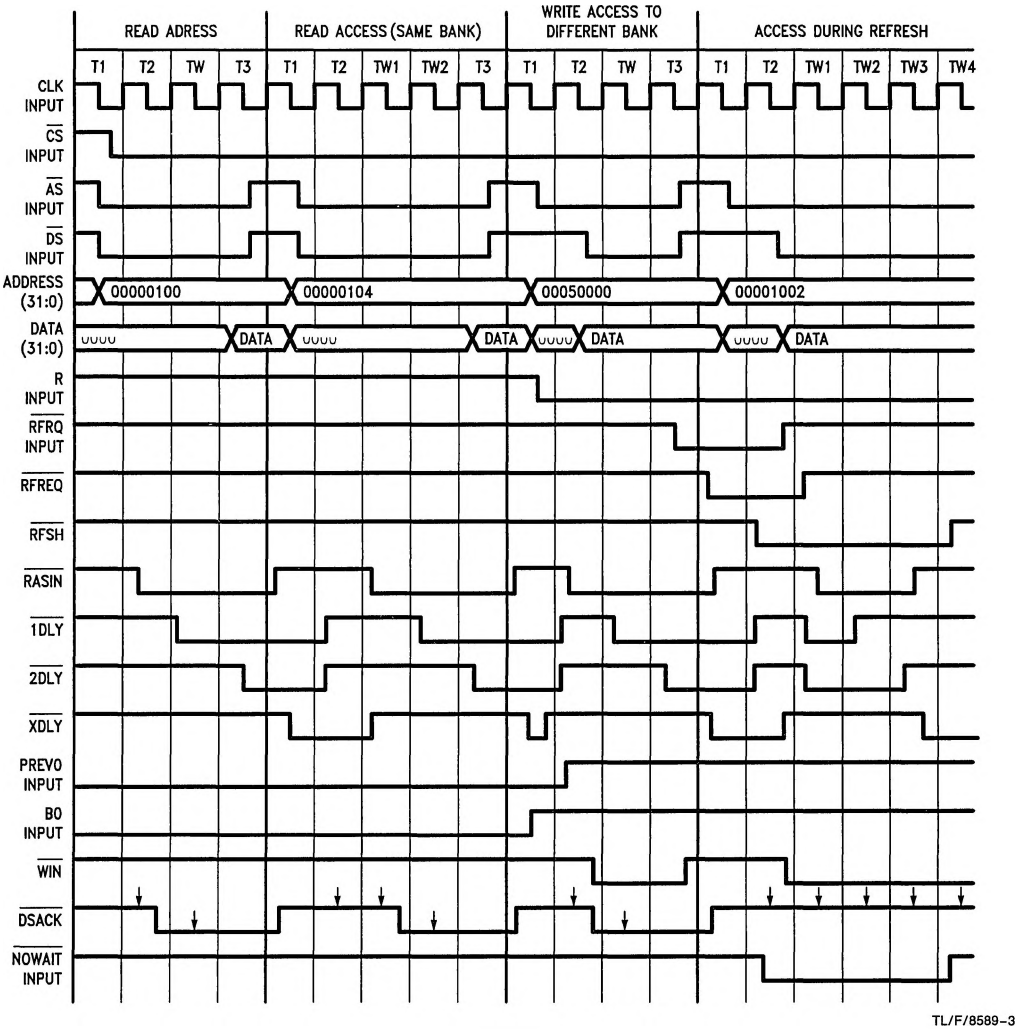
Interface PAL # 1 Between 68020 and DP8418/28 (Continued)



TL/F/8589-2

FIGURE 2. PAL # 1 Simulation Diagram

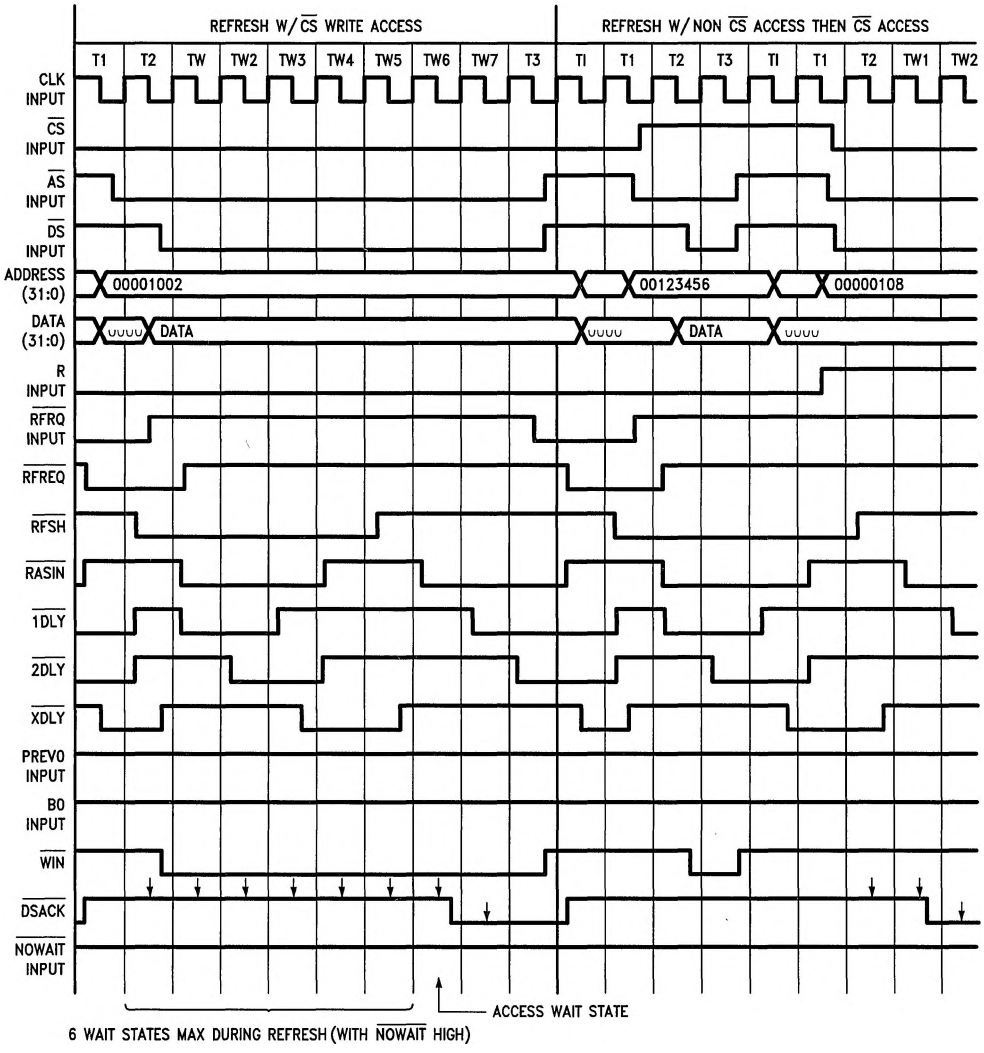
Interface PAL # 1 System Timing Diagram



TL/F/8589–3

FIGURE 3

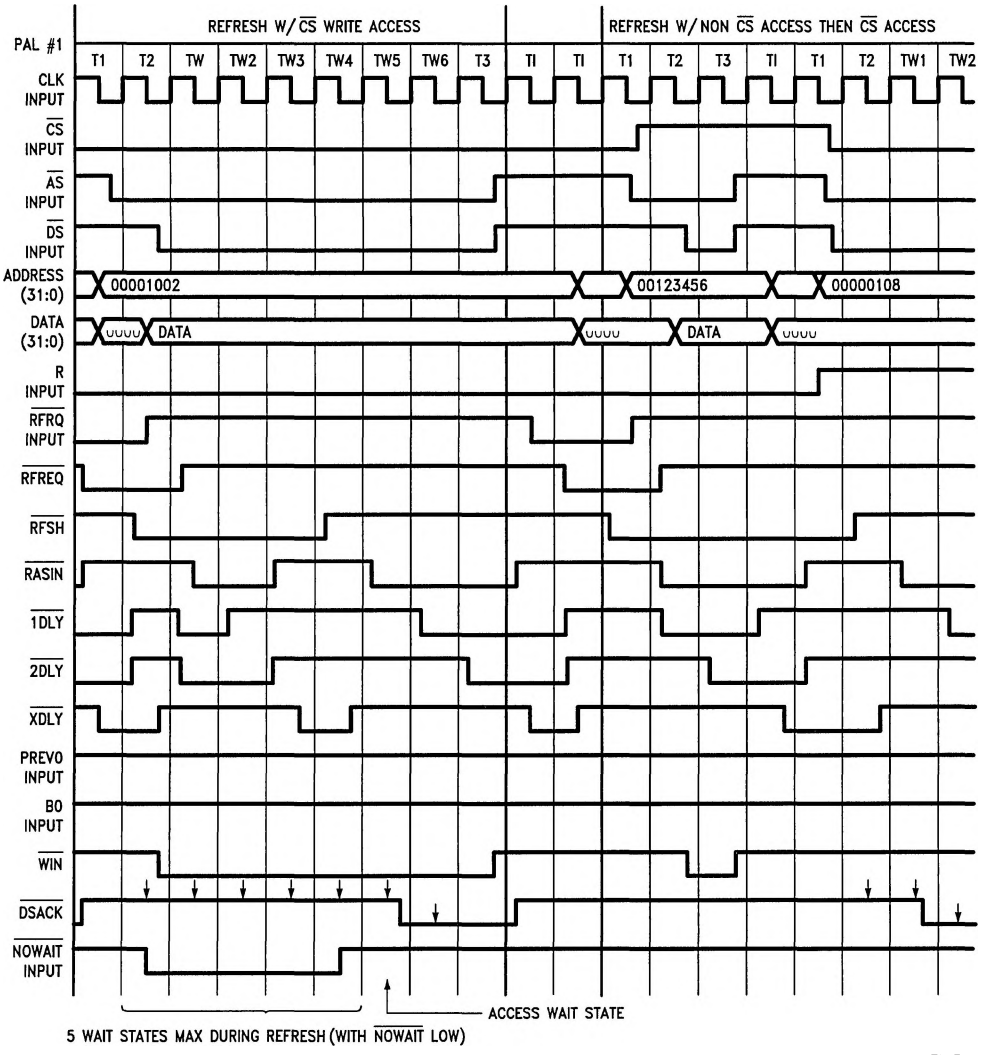
Interface PAL #1 System Timing Diagram (Continued)



TL/F/8589–4

FIGURE 4

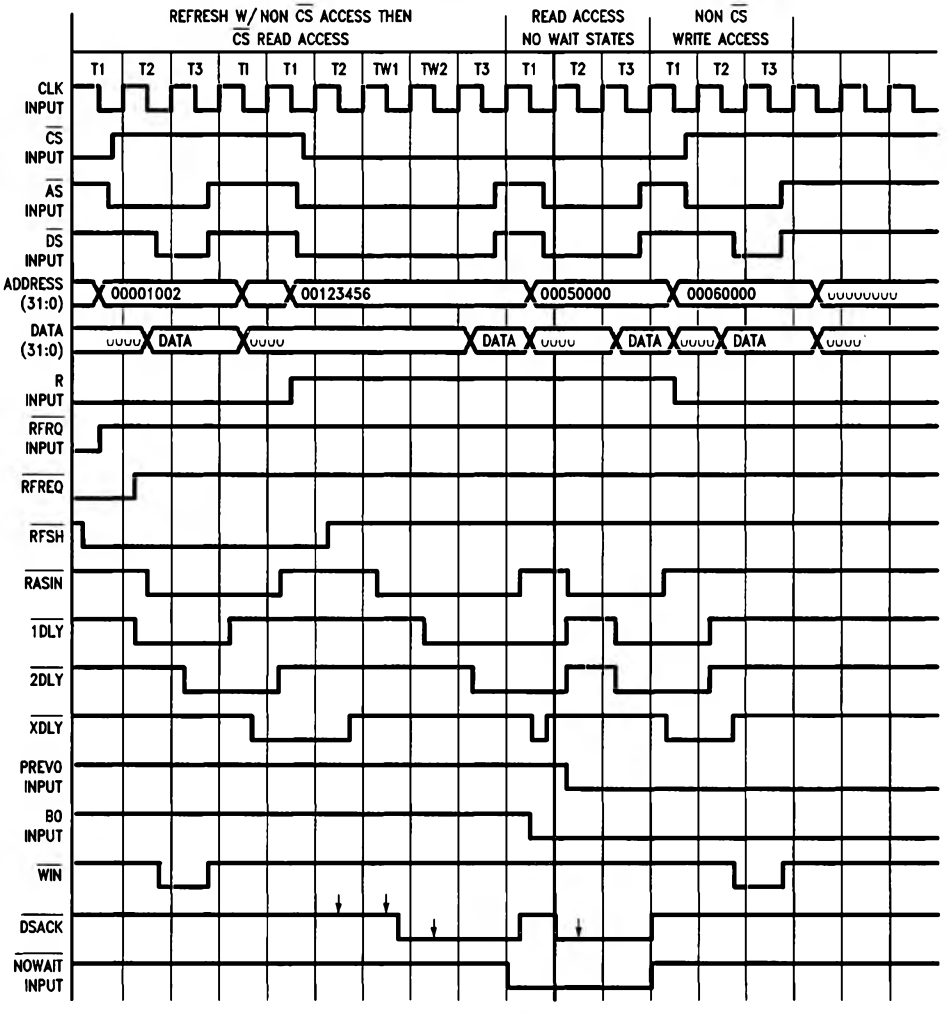
Interface PAL # 1 System Timing Diagram (Continued)



TL/F/8589–5

FIGURE 5

Interface PAL # 1 System Timing Diagram (Continued)



TL/F/8589–6

FIGURE 6

Interface PAL #2 Boolean Equations

This PAL is similar to PAL #1 but ends "$\overline{\text{RASIN}}$" one half period earlier than PAL #1 and relies on the external generation of byte "$\overline{\text{CAS}}$'s" to hold the data valid during 68020 READ access cycles.

```
DMPAL16R4A
CK  /AS  /RFRQ  /CS  R  CLK  NC1  NC2  /NOWAIT  GND
/OE  /DSACK  /RFREQCK  /RFREQ  /2DLY  /1DLY  /RFSH  /RASIN  /XDLY  VCC

IF (VCC) /XDLY = RFSH*/2DLY*/AS                ;"/XDLY" during access
     +/RFSH*1DLY*/2DLY*/RASIN*NOWAIT*/CLK      ;"/XDLY" during RFSH
     +/RFSH*1DLY*2DLY*RASIN*RFREQ*/NOWAIT      ;"/XDLY" during RFSH
     +/RFSH*/XDLY*RFREQ                        ;Hold "/XDLY" low
     +/XDLY*RASIN*/AS*CLK                      ;Hold "/XDLY" low

IF (VCC) /RASIN = /RFSH*RFREQ*/1DLY          ;RFSH "/RASIN"
  +/RFSH*/RASIN*/2DLY*XDLY                   ;Hold in RFSH
  +/RFSH*/RASIN*/CLK                         ;Hold in RFSH
  +RFSH*RFREQCK*/AS*/CS*XDLY*CLK             ;Start "/RASIN"
  +RFSH*RFREQCK*/AS*/CS*2DLY*XDLY*CLK        ;After idle states
  +RFSH*/RASIN*/AS*/CS*XDLY                  ;Hold "/RASIN" low
  +RFSH*/RASIN*CLK                           ;Hold "/RASIN" low

IF (VCC) /RFREQCK = /RFREQ*/CLK              ;Start from falling clock
     +/RFREQCK*/RFREQ                        ;  edge

IF (VCC) /DSACK = /CS*RFSH*/RASIN*NOWAIT*/CLK   ;One WAIT state
  +/DSACK*/CS*RFSH*/RASIN*/AS                   ;Hold "/DSACK" low
  +/CS*RFSH*/AS*/NOWAIT*XDLY                    ;No WAIT state in access

/RFSH := /RFREQ*RASIN*/1DLY*/2DLY            ;Start RFSH
     +/RFREQ*RASIN*1DLY                      ;Start RFSH
     +/RFSH*/RFREQ                           ;Hold RFSH low
     +/RFSH*/RASIN                           ;Hold RFSH low
     +/RFSH*/1DLY                            ;Hold RFSH low

/1DLY := /RFSH*2DLY*/RFREQ                ;Start "/1DLY" during RFSH
     +/RFSH*/1DLY*2DLY*XDLY               ;Continue "/1DLY" during RFSH
     +RFSH*/RASIN                         ;Start "/1DLY" during /RASIN

/2DLY := /RFSH*/1DLY                      ;Start "/2DLY" during RFSH
     +/RFSH*2DLY*/RFREQ*/NOWAIT           ;Shorten RFSH
     +RFSH*/RASIN*/1DLY                   ;Start "/2DLY" during /RASIN
     +RFSH*/RASIN*/NOWAIT                 ;Shorten access

/RFREQ := /RFRQ                           ;Synchronize to system clock
```

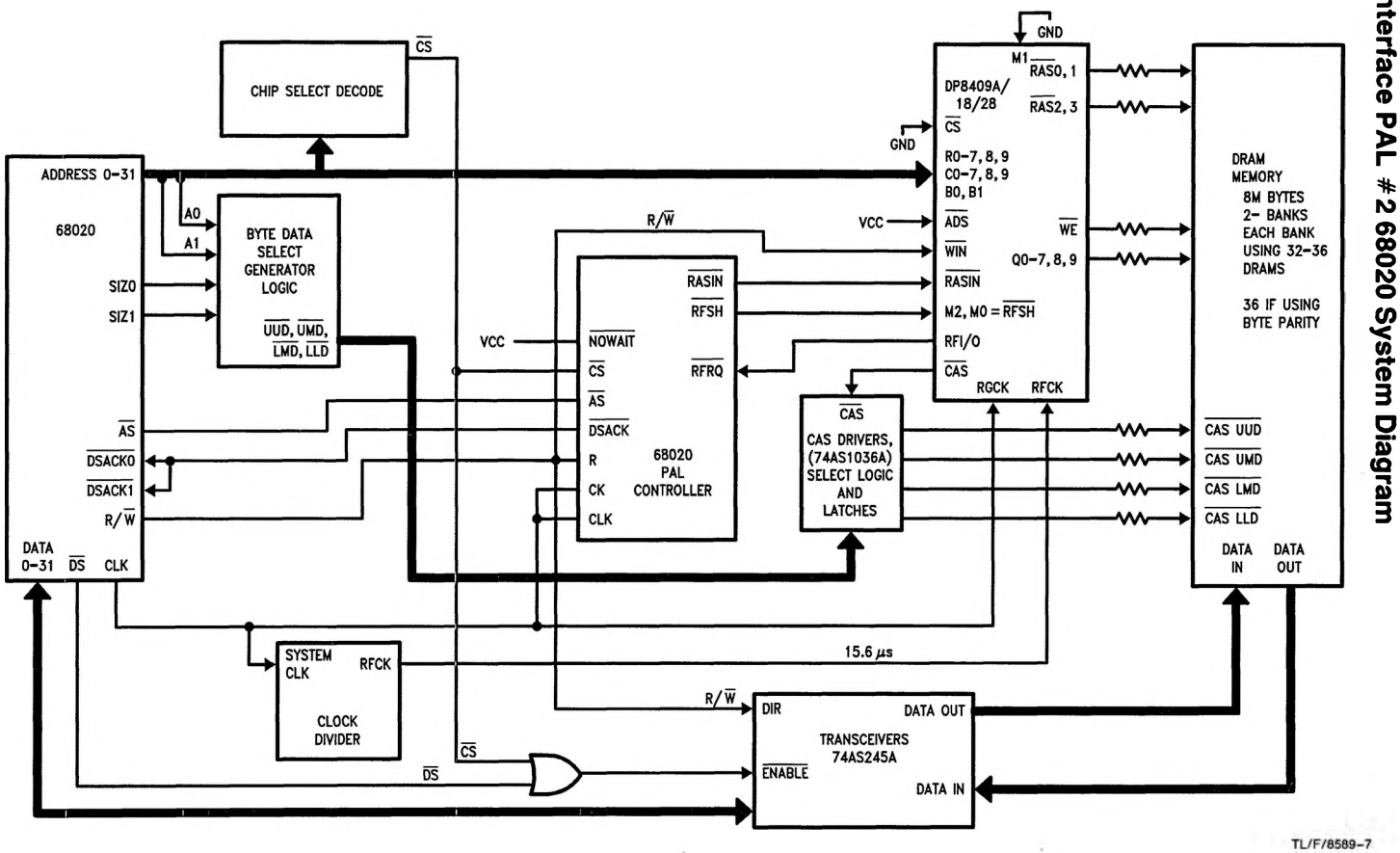
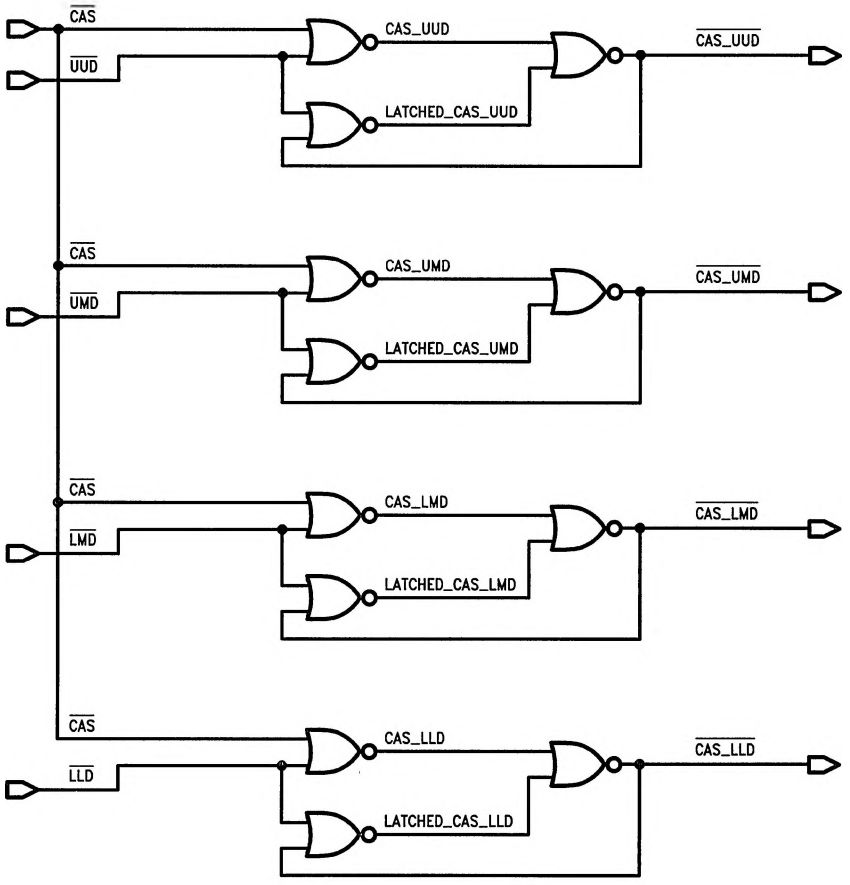

CHIP SELECT DECODE
CS
ADDRESS 0-31
68020
A0
A1
BYTE DATA SELECT GENERATOR LOGIC
UUD, UMD, LMD, LLD
SIZ0
SIZ1
AS
DSACK0
DSACK1
R/W
DATA 0-31
DS
CLK
R/W
VCC
NOWAIT
CS
AS
DSACK
R
CK
CLK
68020 PAL CONTROLLER
RASIN
RFSH
RFRQ
GND
M1
DP8409A/ 18/28
CS
R0-7, 8, 9
C0-7, 8, 9
B0, B1
ADS
WIN
RASIN
M2, M0 = RFSH
RFI/O
CAS
RGCK
RFCK
RAS0, 1
RAS2, 3
WE
Q0-7, 8, 9
CAS
CAS DRIVERS, (74AS1036A) SELECT LOGIC AND LATCHES
DRAM MEMORY
8M BYTES
2- BANKS
EACH BANK
USING 32-36
DRAMS
36 IF USING BYTE PARITY
CAS UUD
CAS UMD
CAS LMD
CAS LLD
DATA IN
DATA OUT
SYSTEM CLK
RFCK
CLOCK DIVIDER
15.6 μs
R/W
DIR
DATA OUT
TRANSCEIVERS 74AS245A
ENABLE
DATA IN
CS
DS
TL/F/8589-7

FIGURE 7

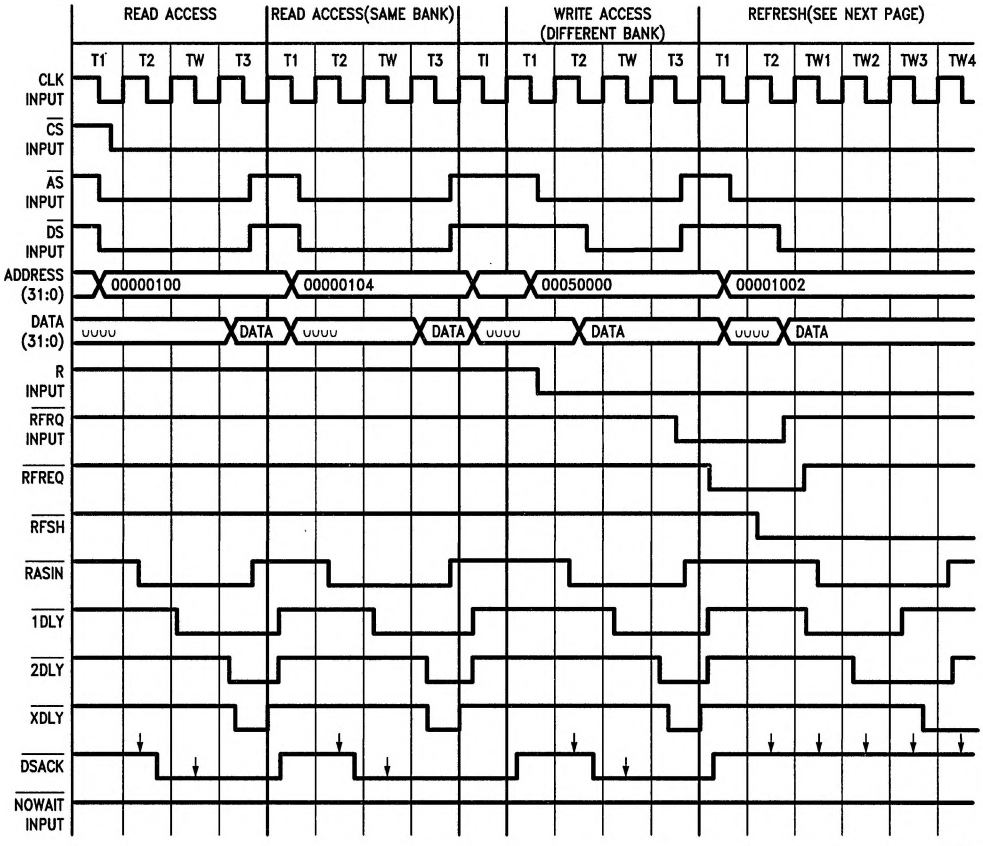
Interface PAL #2 CAS Driver, Select Logic, and Latches



TL/F/8589–8

FIGURE 8

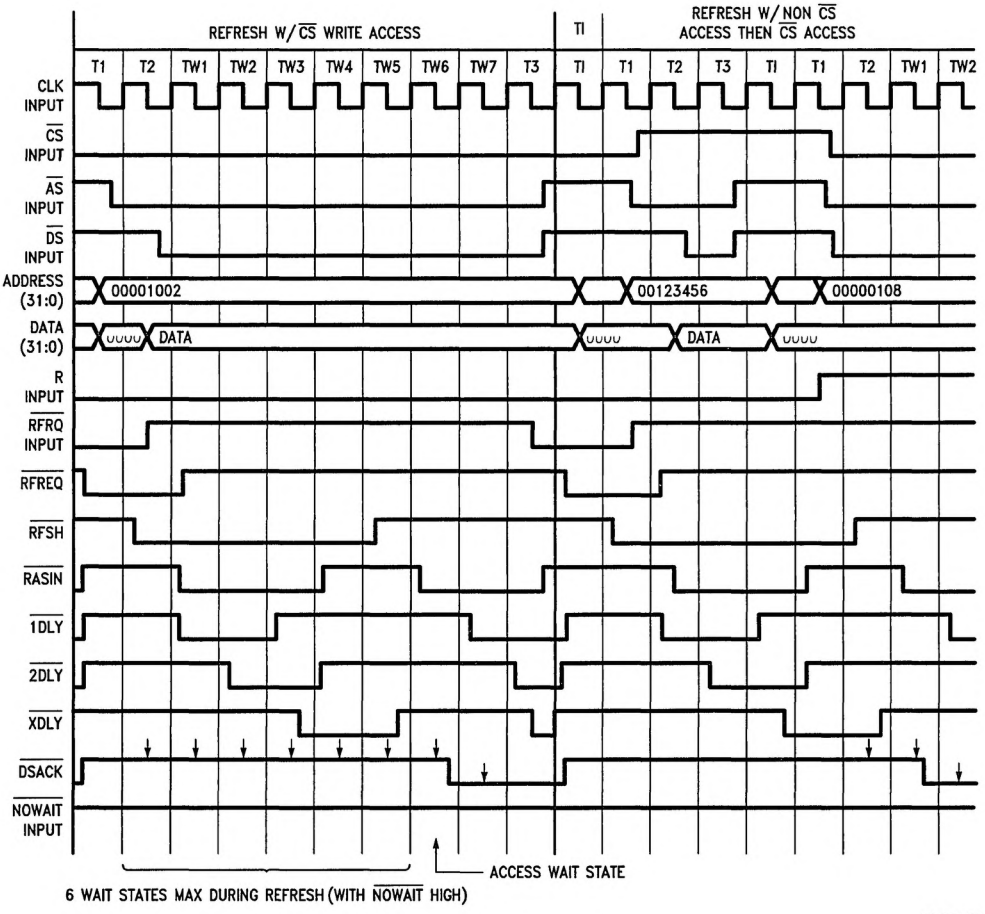
Interface PAL #2 68020 System Timing Diagram



TL/F/8589–9

FIGURE 9

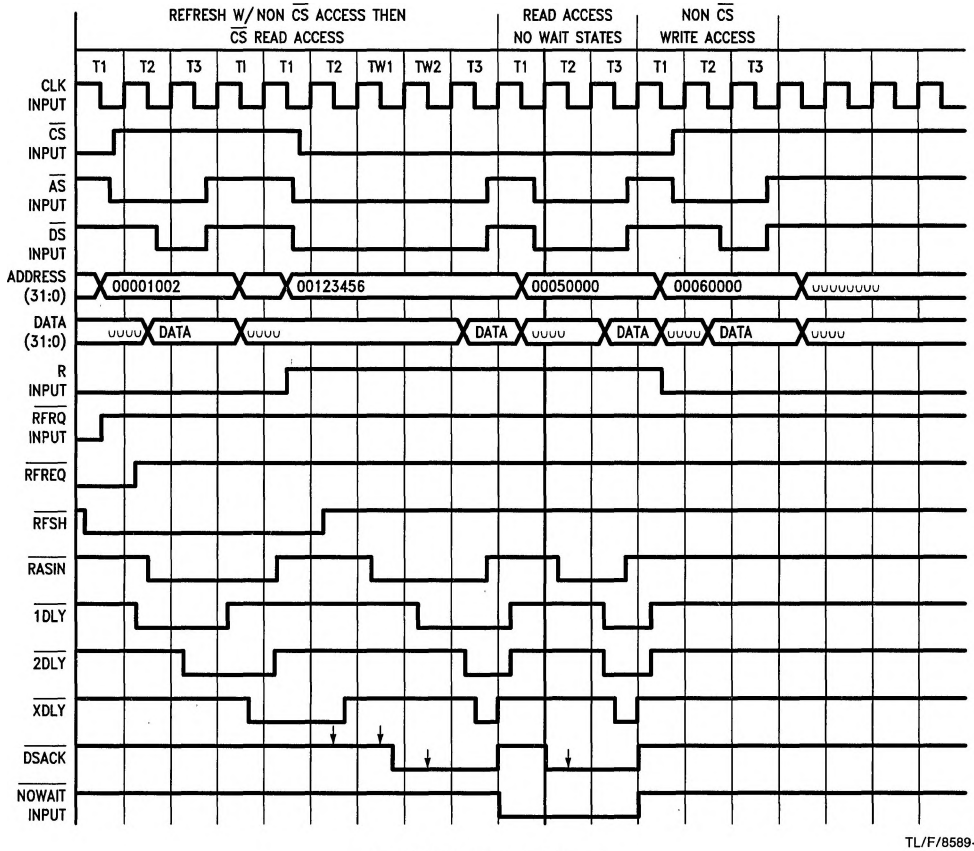
Interface PAL #2 Between 68020 and DP8418



TL/F/8589–10

FIGURE 10. System Timing

Interface PAL #2 Between 68020 and DP8418 (Continued)



TL/F/8589–11

FIGURE 11. System Timing