DATA SHEET

32-Bit to 32/16/8-Bit Dynamic READ/WRITE Bus Sizer

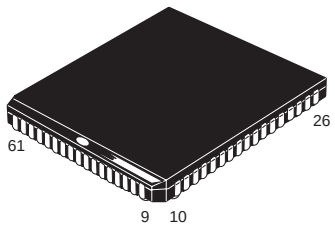
MC68150

NOT RECOMMENDED FOR NEW DESIGNS

The MC68150 Dynamic Bus Sizer is designed to allow the 32-bit MC68/LC/EC040 bus, or other 16- to 32-bit processors, to communicate bi-directionally with 32-, 16-, or 8-bit peripherals and memories. It dynamically recognizes the size of the selected peripheral/memory and then writes or reads the appropriate data to or from that location. Systems designed using the bus sizing feature built into the 68030 can now be easily upgraded to the 68/EC040 by incorporating the MC68150. The 68150 comes in two speed grades: 25/33MHz and 40MHz. These frequencies correspond to their 68040 counterparts. The two grades should be ordered as the MC68150FN33 and MC68150FN40, respectively.

Typical operations which call for bus sizing are booting up instructions from 8-bit ROM (EPROM, EEPROM, etc.) and communicating with 8-bit SRAM's for scratch pad memory storage during interrupt operations. The dynamic property is necessary because the processor does not always know the size of the peripheral it is accessing, as in the case of communicating with a 16-bit VME bus. The MC68150 can also be used to separate a 32-bit "Fast Bus" and an 8-, 16-, or 32-bit "Slow Bus". (See Figure 3)

DYNAMIC READ/WRITE BUS SIZER



FN SUFFIX
PLASTIC PACKAGE
CASE 779-02

EI SUFFIX
Plastic Package
Case 779-02
(Pb-Free Package)

Features

- Allows MC68/LC/EC040 or Other '040 Based Controllers or 68060 to Communicate With 8-Bit Memories and Any MC68XXX Peripheral
- Also Allows Other RISC Processors to Communicate With 8-Bit and 16-Bit Peripherals
- Recognizes the Port (Peripheral) Size Dynamically
- Generates Byte/Word Address to the Dynamic Port
- Generates Byte WRITE Enable Signals For 16- and 8-bit Ports
- Sends a Transfer Acknowledge Signal to the Processor When a Transfer Is Completed
- Synchronization of Data Transfer on Dynamic Port Allows Use of Any Speed Peripheral

1. Overview of Chip Operation

Each access through the MC68150 is started with a chip select ($\overline{CS}$) assertion to the MC68150 - which is generated when a PAL sees a $\overline{TS}$ signal from the '040 - and completed with a transfer acknowledge ($\overline{TA}$) from the MC68150 to the MC68040. The MC68150 has two distinct buses, the MPU bus and the peripheral bus. The MPU bus connects to the processor and includes the transfer control signals (A1, A0, SIZ1, SIZ0, and R/$\overline{W}$), the chip select ($\overline{CS}$), the transfer acknowledge ($\overline{TA}$) and the data bus signals (D31-D0). The peripheral bus consists of the peripheral transfer control signals ($\overline{SWE}$, $\overline{UWE}$, $\overline{LWE}$, $\overline{DS}$, PA1, PA0), and the peripheral transfer acknowledge signals ($\overline{DSACK1}$, $\overline{DSACK0}$) and the peripheral data bus (PD31-PD16).

If a 32-bit peripheral bus is used, then two additional transceivers (e.g. MC74F245) are required for the lower two bytes of the data. These transceivers would be connected to the PD15-PD0 pins on the peripheral side and to the corresponding D15-D0 pins on the MPU bus. The transfer direction is controlled with the R/$\overline{W}$ signal of the processor. The transceivers are enabled only when making an access to a 32-bit port. The D15-D0 pins of the MPU bus on the MC68150 are always disabled until the port size is known, to avoid bus contention when the port is 32-bits.

An access refers to the complete transaction through the MC68150. On the peripheral bus, an access is split into one, two, or four separate transfers.

PIN DESCRIPTIONS

Pin	I/O	Description
BCLK	I	Bus clock — the main system clock (from MC88916)
D31–D0	I/O	Data bits on the 32-bit ‘040 bus
PD31–PD16	I/O	Peripheral data — data bits on the 8/16/32-bit peripheral side; PD15–PD0 do not exist for a 32-bit access, these require two F245 transceivers
PA1, PA0	O	Peripheral address — address bits indicating the byte being accessed on the peripheral side (00 = MSB, 11 = LSB)
$\overline{UWE}$, $\overline{LWE}$	O	Upper (PD31–PD24), lower (PD23–PD16) write enables on the peripheral side significant for 16-bit ports (WRITE)
$\overline{SWE}$	O	Single write enable on the peripheral-bit side for 8-bit ports (WRITE)
$\overline{DS}$	O	Data strobe indicates data can be put on the peripheral bus to be read by the ‘040 (READ) or is valid to be written on the peripheral bus (WRITE)
$\overline{DSACK1}$, $\overline{DSACK0}$	I	Data transfer acknowledge. HH inserts wait states in the current bus cycle. HL indicates the peripheral bus size is 8-bits. LH indicates peripheral bus size is 16-bits. LL indicates the peripheral bus is 32-bits. Recognition of these signals is what allows for “dynamic” bus sizing
SIZ1, SIZ0	I	Indicates the size of the MPU bus access. HH or LL = 32-bits (long word); HL = 16-bits (word); LH = 8-bits (byte)
A1, A0	I	‘040 Address bits, Indicates the byte offset of the ‘040 access (which byte is accessed: HH = D7-D0; HL = D15-D8; LH = D23-D16; LL = D31-D24)
$\overline{CS}$	I	Chip select, tells the 68150 that a transfer is starting or ending
$\overline{TA}$	O	Transfer acknowledge indicates the 68150 has completed the transfer
R/$\overline{W}$	I	READ/WRITE, tells the 68150 if it is a read or write transfer

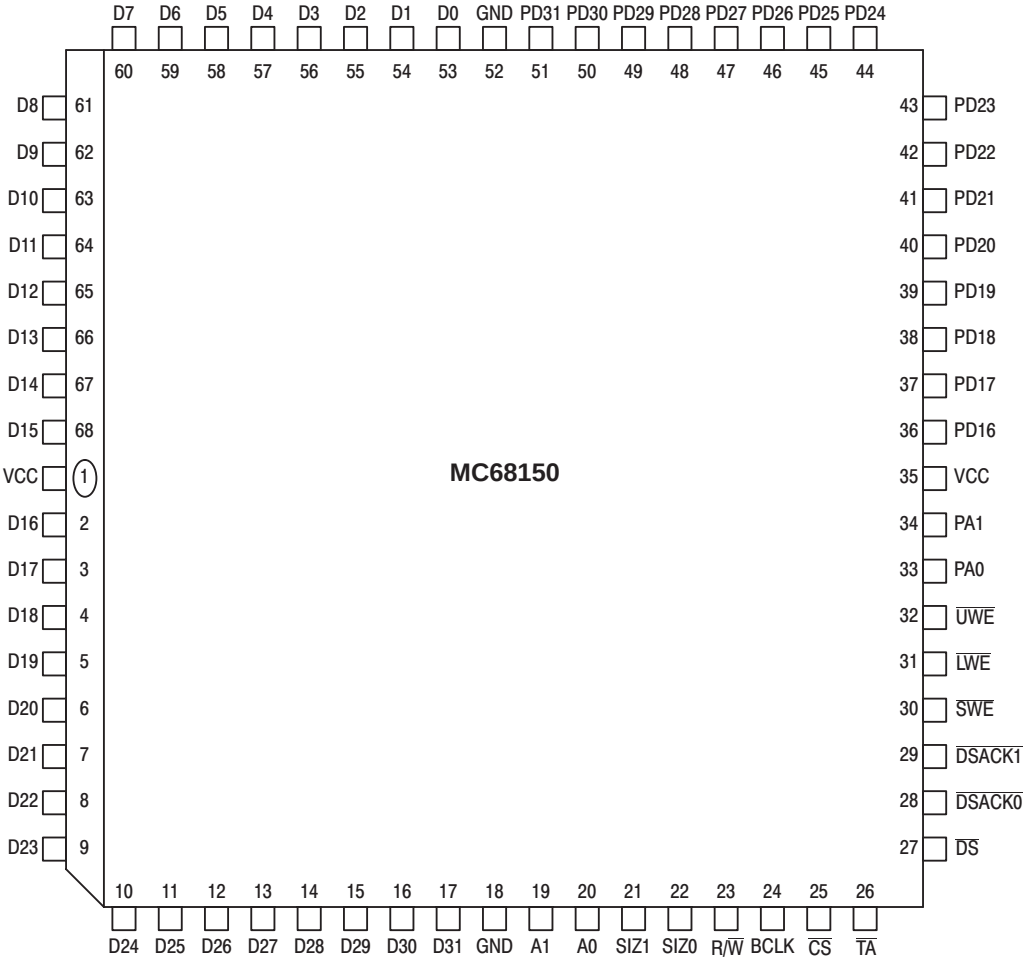


Figure 1. Pinout: 68-Lead PLCC (Top View)

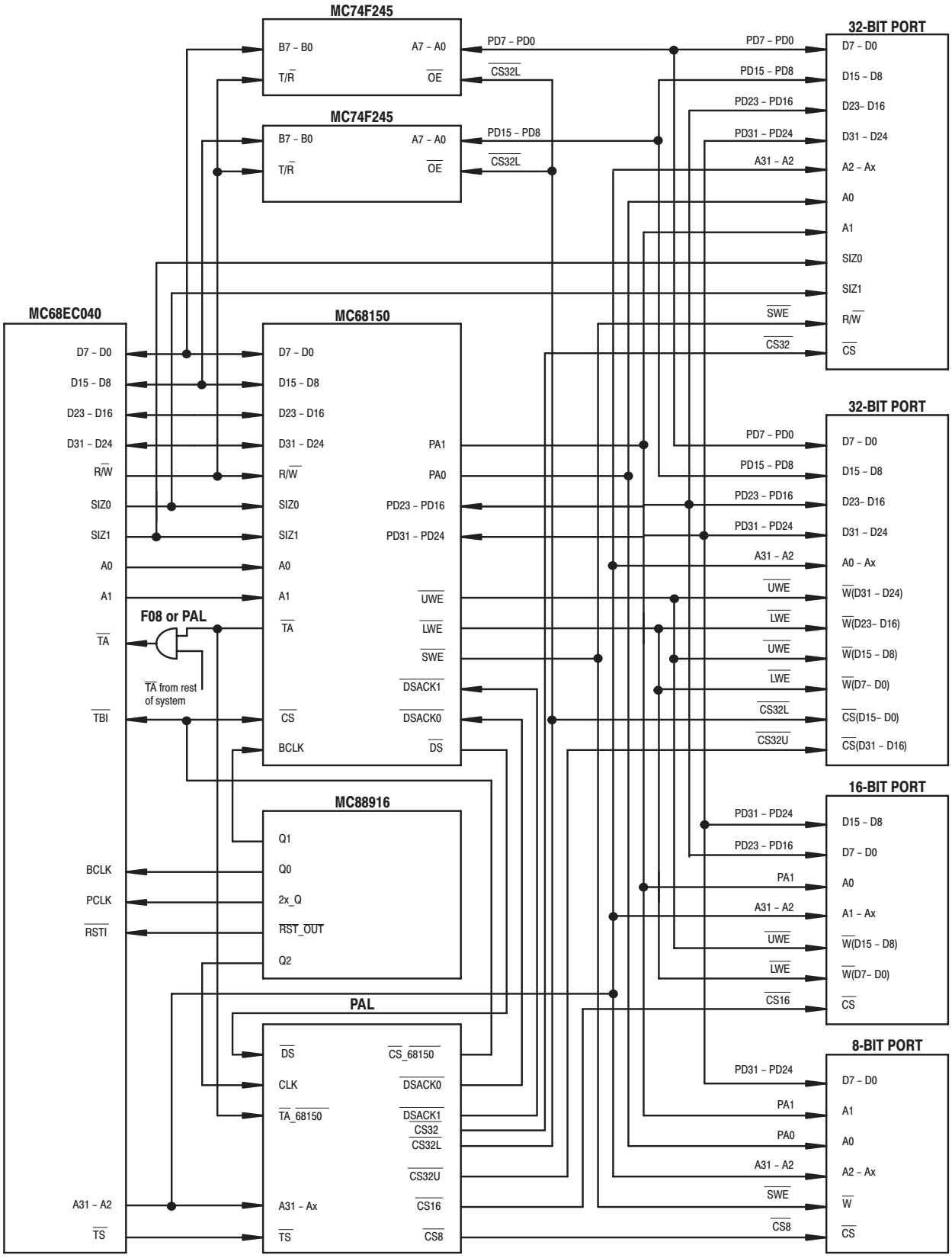


Figure 2. MC68150 Typical System Configuration

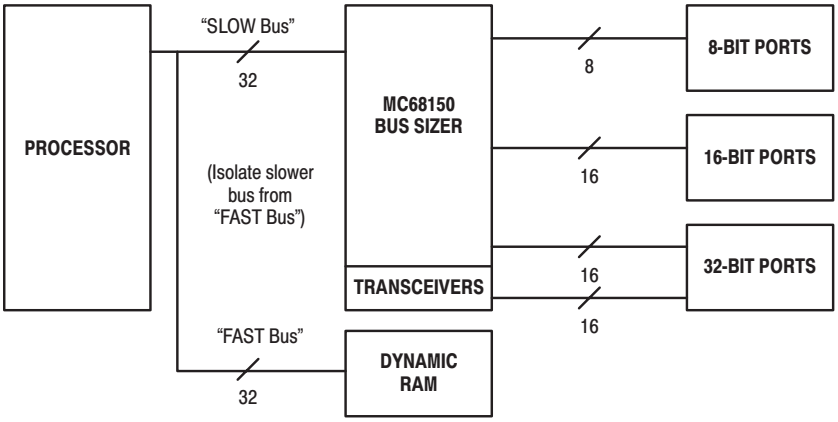


Figure 3. "FAST Bus" "SLOW Bus" Application

Table 1. 68150 TRUTH TABLE (READ Mode)

SIZ1	SIZ0	PA1	PA0	$\overline{DSACK1}$	$\overline{DSACK0}$	D31:24	D23:16	D15:8	D7:0	PD31:24	PD23:16	Description of Transfer	Fig
L	H	L	L	L	L	Out0	X	Z	Z	In0	X	8-Bit READ from 32-Bit Peripheral	49
L	H	L	H	L	L	X	Out1	Z	Z	X	In1		51
L	H	H	L	L	L	X	X	Z	Z	X	X		53
L	H	H	H	L	L	X	X	Z	Z	X	X		55
L	H	L	L	L	H	Out0	X	X	X	In0	X	8-Bit READ from 16-Bit Peripheral	41
L	H	L	H	L	H	X	Out1	X	X	X	In1		43
L	H	H	L	L	H	X	X	Out2	X	In2	X		45
L	H	H	H	L	H	X	X	X	Out3	X	In3		47
L	H	L	L	H	L	Out0	X	X	X	In0	X	8-Bit READ from 8-Bit Peripheral	33
L	H	L	H	H	L	X	Out1	X	X	In1	X		35
L	H	H	L	H	L	X	X	Out2	X	In2	X		37
L	H	H	H	H	L	X	X	X	Out3	In3	X		39
H	L	L	X	L	L	Out0	Out1	Z	Z	In0	In1	16-Bit READ from 32-Bit Peripheral	29
H	L	H	X	L	L	X	X	Z	Z	X	X		31
H	L	L	X	L	H	Out0	Out1	X	X	In0	In1	16-Bit READ from 16-Bit Peripheral	25
H	L	H	X	L	H	X	X	Out2	Out3	In2	In3		27
H	L	L	L	H	L	Out0	X	X	X	In0	X	16-Bit READ from 8-Bit Peripheral	21
H	L	L	H	H	L	↱	Out1	X	X	In1	X	(2 Transfers)	
H	L	H	L	H	L	X	X	Out2	X	In2	X	16-Bit READ from 8-Bit Peripheral	23
H	L	H	H	H	L	X	X	↱	Out3	In3	X	(2 Transfers)	
LL or HH		X	X	L	L	Out0	Out1	Z	Z	In0	In1	32-Bit READ from 32-Bit Peripheral	17
LL or HH		L	X	L	H	Out0	Out1	X	X	In0	In1	32-Bit READ from 16-Bit Peripheral	15
LL or HH		H	X	L	H	↱	↱	Out2	Out3	In2	In3	(2 Transfers)	
LL or HH		L	L	H	L	Out0	X	X	X	In0	X	32-Bit READ from 8-Bit Peripheral	13
LL or HH		L	H	H	L	↱	Out1	X	X	In1	X	(4 Transfers)	
LL or HH		H	L	H	L	↱	↱	Out2	X	In2	X		
LL or HH		H	H	H	L	↱	↱	↱	Out3	In3	X		

X = Don't Care (Do Not Leave Inputs Floating); ↱ = Latched Data From Peripheral; Z = 3-State; InX = Input (0 Signifies the MSB and 3 Signifies the LSB); OutX = Output Corresponding to the Input InX

Table 2. 68150 TRUTH TABLE (WRITE Mode)

SIZ1	SIZ0	PA1	PA0	$\overline{DSACK1}$	$\overline{DSACK0}$	D31:24	D23:16	D15:8	D7:0	PD31:24	PD23:16	Description of Transfer	Fig
L	H	L	L	L	L	In0	X	X	X	Out0	X	8-Bit WRITE to 32-Bit Peripheral	50
L	H	L	H	L	L	X	In1	X	X	X	Out1		52
L	H	H	L	L	L	X	X	In2	X	X	X		54
L	H	H	H	L	L	X	X	X	In3	X	X		56
L	H	L	L	L	H	In0	X	X	X	Out0	X	8-Bit WRITE to 16-Bit Peripheral	42
L	H	L	H	L	H	X	In1	X	X	X	Out1		44
L	H	H	L	L	H	X	X	In2	X	Out2	X		46
L	H	H	H	L	H	X	X	X	In3	X	Out3		48
L	H	L	L	H	L	In0	X	X	X	Out0	X	8-Bit WRITE to 8-Bit Peripheral	34
L	H	L	H	H	L	X	In1	X	X	Out1	X		36
L	H	H	L	H	L	X	X	In2	X	Out2	X		38
L	H	H	H	H	L	X	X	X	In3	Out3	X		40
H	L	L	X	L	L	In0	In1	X	X	Out0	Out1	16-Bit WRITE to 32-Bit Peripheral	30
H	L	H	X	L	L	X	X	In2	In3	X	X		32
H	L	L	X	L	H	In0	In1	X	X	Out0	Out1	16-Bit WRITE to 16-Bit Peripheral	26
H	L	H	X	L	H	X	X	In2	In3	Out2	Out3		28
H	L	L	L	H	L	In0	In1	X	X	Out0	X	16-Bit WRITE to 8-Bit Peripheral	22
H	L	L	H	H	L	↱	↱	X	X	Out1	X	(2 Transfers)	
H	L	H	L	H	L	X	X	In2	In3	Out2	X	16-Bit WRITE to 8-Bit Peripheral	24
H	L	H	H	H	L	X	X	↱	↱	Out3	X	(2 Transfers)	
LL or HH		X	X	L	L	In0	In1	In2	In3	Out0	Out1	32-Bit WRITE to 32-Bit Peripheral	18
LL or HH		L	X	L	H	In0	In1	In2	In3	Out0	Out1	32-Bit WRITE to 16-Bit Peripheral	16
LL or HH		H	X	L	H	↱	↱	↱	↱	Out2	Out3	(2 Transfers)	
LL or HH		L	L	H	L	In0	In1	In2	In3	Out0	X	32-Bit WRITE to 8-Bit Peripheral	14
LL or HH		L	H	H	L	↱	↱	↱	↱	Out1	X	(4 Transfers)	
LL or HH		H	L	H	L	↱	↱	↱	↱	Out2	X		
LL or HH		H	H	H	L	↱	↱	↱	↱	Out3	X		

X = Don't Care (Do Not Leave Inputs Floating); ↱ = Latched Data From Processor; Z = 3-State; InX = Input (0 Signifies the MSB and 3 Signifies the LSB); OutX = Output Corresponding to the Input InX

TYPICAL SYSTEM CONFIGURATION

A 68040 system using the MC68150 for dynamic bus sizing consists minimally of the MC68040 microprocessor, the MC68150, a PAL for timing control, and various size memory devices. When a 32-bit port is required, two F245's must be used to buffer the PD15–PD0 bits to the D15–D0 bits of the '040, since the MC68150 does not support these bits. The typical system configuration (Figure 2) illustrates communication between a 68040 and 8-, 16-, and 32-bit peripheral chips.

Design Guidelines

- Peripheral data connections are the same as for a 68030
  - Connect PD31:24 for 8-bit port
  - Connect PD31:24 for upper byte 16-bit port
  - Connect PD23:16 for lower byte 16-bit port
  - Connect PD31:24 for upper-upper byte 32-bit port
  - Connect PD23:16 for upper-middle byte 32-bit port
  - Connect PD15:8 for lower-middle byte 32-bit port
  - Connect PD7:0 for lower-lower byte 32-bit port
- 32-bit ports require transceivers or latched transceivers for PD15:0
  - The MC68150 only passes the upper 16 data bits on a 32-bit transfer
- 68150 does not support burst mode
  - One way to handle this is to connect $\overline{TA}$ of the 68150 to $\overline{TBI}$ of the 68040

2. MC68040 BUS OPERATION

An access is divided into multiple states. Each state represents half a clock period. All even states are defined when BCLK is High, all odd states are defined when BCLK is low. A clock edge is referenced by the state that follows the clock edge. All rising edges are referenced by even number states. All falling edges are referenced by odd number states. DS0 is the first state of an access. DSW represents a wait state or a mid-access transfer state. A wait state indicates an access is occurring, but that the MC68150 is waiting on the peripheral to complete the transaction. Note that the peripheral and MPU states are distinct, though related, to each other. The peripheral states start with S0. Figure 4 is an example transfer with the states marked.

2.1 Access Start

The MC68040 begins an access by asserting the transfer control signals and transfer start ($\overline{TS}$). The transfer control signals are held by the MC68040 throughout the access. The transfer start is asserted around only one rising edge of the clock (BCLK). The chip select ($\overline{CS}$) for the MC68150 is asserted while the MC68040 transfer control signals are input to the MC68150.

The transfer control signals (A1, A0, SIZ1, SIZ0, R/$\overline{W}$) must all be valid a set-up time before the rising edge of BCLK on which $\overline{CS}$ recognized (DS2). If the transfer control signals change states during this set-up time, the MC68150 operation is unpredictable. The transfer control signals must be held valid until at least DS4. If the $\overline{CS}$ switches during the set-up time before the rising edge of DS2, then the access may not be recognized until the next clock edge. Once asserted, $\overline{CS}$ must be held asserted until the end of the access.

During a write access, the data signals (D(31:0)) must be valid a set-up time before the rising edge of DS2. The ‘040 data is latched into the 68150 off the DS2 rising clock edge (as long as $\overline{CS}$ is recognized as described above). This latched data is internally held by the 68150 until $\overline{CS}$ is negated at the end of a transfer.

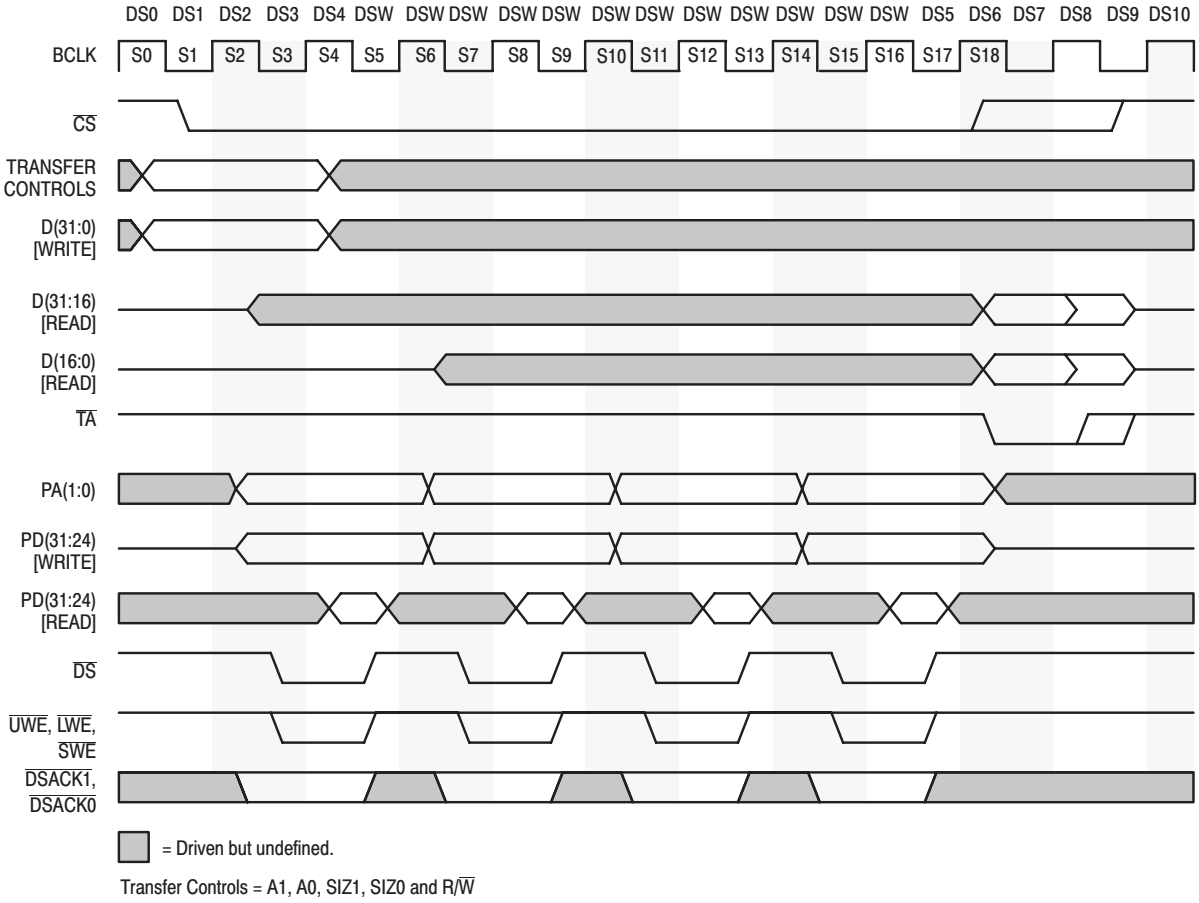


Figure 4. MC68150 32-Bit to 8-Bit Transfer Example (READ or WRITE)

2.2 Early Access Termination

An access through an MC68150 can be terminated before completion by negating $\overline{CS}$ early. The $\overline{CS}$ negation is recognized on a rising edge of BCLK. The $\overline{CS}$ **early negation** is used when the access should be ignored, such as when a bus error occurs. Any data transferred through the MC68150 is lost when an early access negation occurs. If the access must be completed at a later time, the entire access must be repeated. To guarantee that $\overline{TA}$ is not asserted during early negation, $\overline{CS}$ must be negated before DS6.

For peripherals that can be read or written to twice, special care must be taken in aborting the 68150 access.

2.3 Early Release of the MPU Bus

Though early MPU bus release is not economical for most applications, it may allow an incremental improvement in performance at the expense of additional logic.

The transfer control signals (A1, A0, SIZ1, SIZ0, R/$\overline{W}$) are held valid at least until DS4. During a write access, the data signals are held valid at least until DS4. After this hold time, the transfer control signals may change without affecting operation as long as $\overline{CS}$ remains asserted. This allows the MC68150 to release the MPU bus before the access is complete on the peripheral side of the MC68150. During a read access, early release is of limited use, because the MC68040 will not be able to use the bus until the peripheral data has been read. On a write, early MPU bus release does allow the MC68040 to continue with the next operation while the MC68150 completes the access.

When using this early release feature, a bus error could be difficult to handle. If the peripheral asserts a bus error after the MC68040 receives a $\overline{TA}$, meaning the transfer is complete, then a subsequent bus error assertion to the MC68040 will not match the offending address with the bus error address. This can be handled by software or hardware that reads a bit to see if the bus error is coming from the MC68150.

Another consideration in using early bus release is the handling of back to back transfers and transfer acknowledges. The chip select logic for the MC68150 must recognize when a second access occurs while completing an access on the peripheral side. The $\overline{CS}$ must be negated for one rising edge of BCLK between accesses. The transfer acknowledge from the MC68150 signals the end of the access. If early bus release is used, then the transfer acknowledge generated by the MC68150 should not be sent to the MC68040.

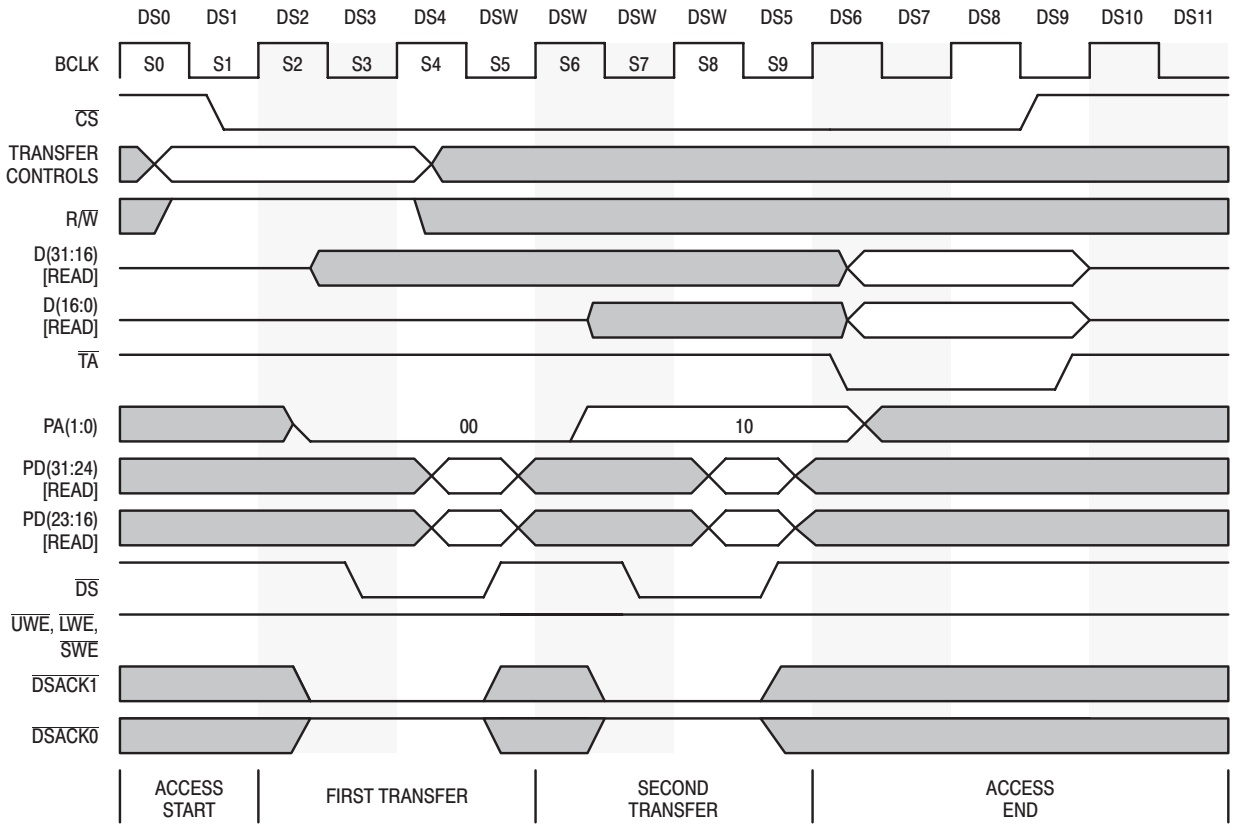


Figure 5. 32-Bit '040 READ from 16-Bit Peripheral With Asynchronous Termination

2.4 Access Termination

The final transfer on the peripheral side of the MC68150 occurs during DS5. The MC68150 access is normally terminated by the MC68150 asserting $\overline{TA}$ during DS6. Once the $\overline{TA}$ is asserted, $\overline{TA}$ is held asserted until at least DS8. The MC68150 negates $\overline{TA}$ **asynchronously** if $\overline{CS}$ is negated after DS7; an example of asynchronous termination is shown in Figure 5. An abnormal MC68150 termination occurs when the $\overline{CS}$ is negated before DS6. See early access termination for more details.

For a normal access, $\overline{CS}$ is held asserted until at least DS6. $\overline{TA}$ negation is synchronous to the rising edge of DS8 if $\overline{CS}$ meets a set-up to a DS8 rising BCLK edge. Asynchronous termination is defined as any $\overline{CS}$ negation beyond the DS8 rising BCLK edge. In this case the $\overline{TA}$ occurs a delay after the $\overline{CS}$ negation, because it is asynchronous termination.

During a read access, the data being presented to the MC68040 is held for at least one rising edge of BCLK (DS8). When $\overline{CS}$ is negated, the MPU data bus goes to high impedance. If the data must be held longer than a single period, then asynchronous $\overline{TA}$ negation is necessary.

2.5 MPU Bus Idle State

$\overline{CS}$ must remain high for one rising edge of BCLK between each access. This is guaranteed by the MC68040 if $\overline{CS}$ is asserted on the clock edge following $\overline{TS}$ assertion (or later) on all accesses through the MC68150. When the MPU bus is idle (between accesses), the D(31:0) are in high impedance and $\overline{TA}$ is negated.

3. PERIPHERAL BUS OPERATION

The peripheral access is divided into one or more transfers. The transfers are divided into multiple states. Each state represents half a clock period. All even states are the BCLK HIGH half of the clock period. All odd states are the BCLK LOW half of the clock period. A clock edge is referenced by the state that follows the clock edge. All rising edges are referenced by even number states. All falling edges are referenced by odd number states. S0 is the first state of a peripheral access. Note that the peripheral and MPU states are distinct, though related to each other.

3.1 Initial Transfer Start

A transfer starts off the rising edge of BCLK following the $\overline{CS}$ assertion (S2). The starting address of the transfer (PA1, PA0) is asserted during S2. The MC68040 A(31:2) are routed past the MC68150, directly to the peripheral bus. Buffers can be used to minimize loading on the MC68040 address bus. If the access is a write, the peripheral data bus (PD(31:16)) is driven during S2.

The starting data strobe ($\overline{DS}$) is asserted during S3. If the access is a write, the starting write enables ($\overline{SWE}$, $\overline{UWE}$, and $\overline{LWE}$) are also asserted during S3. $\overline{SWE}$ is the write enable for 8-bit peripherals. The $\overline{SWE}$ is asserted for every transfer of a write access. It indicates the data on PD(31:24) is valid and ready to be written to the 8-bit peripheral. $\overline{UWE}$ and $\overline{LWE}$ are the write enables for 16-bit peripherals. $\overline{UWE}$ indicates the data on PD(31:24) is valid and ready to be written to the 16-bit peripheral. $\overline{LWE}$ indicates the data on PD(23:16) is valid and ready to be written to the 16-bit peripheral.

A 32-bit port can generate write enables from the MC68040 normally (A1, A0, SIZ1, SIZ0 and R/$\overline{W}$). These write enables can be qualified with $\overline{SWE}$ or $\overline{DS}$ for the 32-bit peripheral. An alternative is to use the $\overline{UWE}$ and $\overline{LWE}$ in conjunction with the PA1 and PA0. If it is a 32-bit access (SIZ1 and SIZ0 = HH or LL), then apply $\overline{UWE}$ to PD(31:24), PD(15:8) and $\overline{LWE}$ to PD(23:16), PD(7:0). If it is a 16-bit access (SIZ1 and SIZ0 = HL), then apply $\overline{UWE}$ to PD(31:24) and $\overline{LWE}$ to PD(23:16).

3.2 Initial Transfer Termination

Starting with S2, the MC68150 monitors the data strobe acknowledge signals ($\overline{DSACK1}$, $\overline{DSACK0}$) for the access termination.

Wait states are inserted into the access by maintaining $\overline{DSACK1}$ and $\overline{DSACK0}$ HIGH. The $\overline{DSACK1}$ and $\overline{DSACK0}$ are sampled on the falling edge of BCLK. $\overline{DSACK1}$ and $\overline{DSACK0}$ must remain valid for two successive falling edges of BCLK to be recognized. If the access is a read, the PD(31:16) must be valid a set-up before the falling edge between S4 and S5. The PD(31:16) must be held valid a hold time into S5. The $\overline{DSACK1}$ and $\overline{DSACK0}$ are first sampled during S2. The $\overline{DSACK1}$ and $\overline{DSACK0}$ are then continuously sampled each falling edge of BCLK until the entire access is completed. This means the fastest initial transfer is two clocks (starting S2 and ending S4).

The size of the peripheral port is indicated by the $\overline{DSACK1}$ and $\overline{DSACK0}$. The port size must be indicated on the first transfer and is the port size for the remainder of the access. Once the peripheral port size is indicated on the first transfer, all subsequent transfers should be terminated with the same port size indication. If another port size is indicated on subsequent transfers of the same access, the MC68150 operation is not guaranteed nor predictable. Depending upon the conditions, the MC68150 may either

ignore the port size change on subsequent transfers or accept the subsequent transfers as a transfer of the original port size. If the MC68150 ignored the improper transfer, the $\overline{DS}$ is not negated. If the MC68150 accepted the improper transfer as an original port size transfer, $\overline{DS}$ negates normally. If a different port size is indicated on subsequent transfers, the changing of $\overline{DSACK1}$ and $\overline{DSACK0}$ states on subsequent transfers may or may not insert wait states into the access.

When a valid $\overline{DSACK1}$ and $\overline{DSACK0}$ state has been held for two consecutive falling edges of BCLK, the intial transfer is ended (falling edges of S3 and S5). $\overline{DS}$ is negated during S5. If the access is a write, $\overline{SWE}$, $\overline{UWE}$ and $\overline{LWE}$ are all negated during S5. PA1 and PA0 are held valid until S6. If the access is a write, PD(31:16) are held valid until S6.

Table 3. $\overline{DSACK1}$ AND $\overline{DSACK0}$ DECODE LOGIC

$\overline{CS}$	$\overline{DSACK1}$	$\overline{DSACK0}$	
H	X	X	No Access
L	L	L	Complete 32-Bit Access
L	L	H	Complete 16-Bit Access
L	H	L	Complete 8-Bit Access
L	H	H	Insert Wait States

3.3 Subsequent Access Termination

If additional transfers are required to complete an access (e.g. a long word access to a byte port requires four transfers to complete), then PA1 and PA0 are changed during S6. If the access is a write, PD(31:16) is changed during S6. $\overline{DS}$, $\overline{SWE}$, $\overline{UWE}$ and $\overline{LWE}$ are asserted during S7. The $\overline{DSACK1}$ and $\overline{DSACK0}$ are sampled again for the subsequent transfer during S6.

If the access is a read, the PD(31:16) must be valid a set-up before the falling edge between S8 and S9. The PD(31:16) must be held valid a hold time into S9. $\overline{DS}$, $\overline{SWE}$, $\overline{UWE}$ and $\overline{LWE}$ are negated during S9.

Subsequent transfers continue in the same manner with S10 and S14 replacing S6 above, S11 and S15 replacing S7 above, S13 and S17 replacing S9 above, and S14 and S18 replacing S10. Figures 13 and 14 are examples of 32-bit '040 access from/to an 8-bit port, read and write respectively. Both require four sequential data transfers to complete the access.

On a write, the PD(31:16) bus goes into high impedance when $\overline{TA}$ asserts (DS6). To hold the data valid on the PD(31:16) bus beyond the normal access end, wait states must be inserted using $\overline{DSACK1}$, $\overline{DSACK0}$ = HH on the last transfer. The wait states will delay $\overline{TA}$ assertion.

3.4 Peripheral Bus Idle State

The MC68150 enters an idle state on the peripheral bus on the rising edge of BCLK after the last transfer is terminated.

When the peripheral bus is idle, $\overline{DS}$, $\overline{UWE}$, $\overline{LWE}$ and $\overline{SWE}$ are all negated. PA1 and PA0 are driven, but undefined. PD(31:16) are in high impedance. $\overline{DSACK1}$ and $\overline{DSACK0}$ are ignored.

3.5 Peripheral Bus Asynchronous Operation

(Only for operation independant of BCLK non–680X0 processors. See AC Specifications 22–34 in lieu of specs 1–21)

The MC68150 peripheral bus has both asynchronous and sychronous operations. The asynchronous operation allows the user to operate the peripheral bus independently of the BCLK. The synchronous operation requires close attention to the BCLK set-up and hold times.

An asynchronous transfer on the MC68150 begins with the assertion of $\overline{DS}$ by the MC68150. If the access is a write, $\overline{SWE}$, $\overline{UWE}$ and $\overline{LWE}$ are asserted at the same time as $\overline{DS}$. PA1 and PA0 are valid before the $\overline{DS}$ assertion. PD(31:16) are driven and valid before $\overline{DS}$ assertion. To avoid bus contention on the peripheral bus, the R/$\overline{W}$ from the MPU can be used.

The $\overline{DSACK1}$ and $\overline{DSACK0}$ assertions must be within the asynchronous skew specification. If one of the $\overline{DSACK}$ signals is asserted on one falling edge, the asynchronous skew specification ensures that the other $\overline{DSACK}$ signal is valid before the next falling edge. Because the $\overline{DSACK}$ signals can be changing on a falling BCLK edge, an additional wait state may be incurred. $\overline{DSACK1}$ and $\overline{DSACK0}$ can be negated when $\overline{DS}$ is negated.

If the access is a write, the $\overline{UWE}$, $\overline{LWE}$ and $\overline{SWE}$ negate at the same time as $\overline{DS}$. PD(31:16) change state after the $\overline{DS}$, $\overline{UWE}$, $\overline{LWE}$ and $\overline{SWE}$ are negated.

If the access is a read, PD(31:0) is valid after $\overline{DSACK1}$ and $\overline{DSACK0}$ are asserted and held until $\overline{DS}$ is negated.

MAXIMUM RATINGS*

Symbol	Parameter	Value	Unit
V_{CC}	DC Supply Voltage (Referenced to GND)	–0.3 to +7.0	V
V_{IN}	DC Input Voltage (Referenced to GND)	–0.5 to +7.0	V
V_{OUT}	DC Output Voltage (Referenced to GND)	–0.5 to V_{CC} + 0.5	V
I_{IN}	DC Input Current, Per Pin	±20	mA
I_{OUT}	DC Output Sink/Source Current, Per Pin	±35	mA
T_{STG}	Storage Temperature Range	–55 to +150	°C
T_A	Operating Ambient Temperature Range	–40 to +85	°C

* Maximum Ratings are those values beyond which damage to the device may occur. Functional operation should be restricted to the Recommended Operating Conditions.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
V_{CC}	DC Supply Voltage (Referenced to GND)	4.5	5.5	V
V_{IN}, V_{OUT}	DC Input Voltage, Output Voltage (Referenced to GND)	0	V_{CC}	V
T_A	Operating Ambient Temperature Range	–40	85	°C
I_{OL}/I_{OH}	Output Current, LOW/HIGH (Control Pins) (Data Pins)		±16 ±5	mA

OTHER CHARACTERISTICS

Symbol	Parameter	Conditions	Ratings	Unit
C_{IN}	Input Capacitance	V_I = 0V or V_{CC}	8.5	pF
C_{OUT}	Output Capacitnace	V_I = 0V or V_{CC}		pF
$C_{I/O}$	Input/Output Capacitance	V_I = 0V or V_{CC}		pF
$I_{LATCHUP}$	Latchup Current		>500	mA
V_{ESD}	Electrostatic Discharge Voltage		8,000	V
P_D	Power Dissipation in Still Air (Calculated at Worst Case)		530	mW

NOTE: Rating values are design targets. Actual performance will be noted upon completion of characterization.

I_{CC} versus FREQUENCY

Frequency (MHz)	No Load (50pF Only) 25°C I_{CC} (mA)			Loaded (50pF/500Ω) 25°C I_{CC} (mA)		
	4.5	5.0	5.5	4.5	5.0	5.5
1	2	4	6	65	73	82
10	13	16	20	75	83	93
20	25	29	35	85	95	106
30	38	43	50	95	106	120
40	51	59	67	105	120	135

DC SPECIFICATIONS

Symbol	Characteristic	Min	Max	Unit	Condition
V_{IH}	Input HIGH Voltage	2.0	V_{CC}	V	V_{CC} = 5V
V_{IL}	Input LOW Voltage	GND	0.8	V	V_{CC} = 5V
I_{IN}	Input Leakage Current SIZ1, SIZ0, A1, A0, $\overline{CS}$, $\overline{DSACK1}$, $\overline{DSACK0}$, R/$\overline{W}$, BCLK		±20	µA	V_{IN} = 0.5V/2.4V, V_{CC} = 5.25V
I_{OZ}	Hi-Impedance (Off-State) Leakage Current D(31:0), PD(31:16)		±20	µA	V_{IN} = V_{IH}/V_{IL} V_O = GND/V_{CC}
ΔICCT (BCLK)	Additional Maximum ICC/Input BCLK Only		7.5	mA	V_{IH} = V_{CC} –2.1V, V_{CC} = 5.25V
ΔICCT	Additional Maximum ICC/Input SIZ1, SIZ0, A1, A0, $\overline{CS}$, $\overline{DSACK1}$, $\overline{DSACK0}$, R/$\overline{W}$		1.5	mA	V_{IH} = V_{CC} –2.1V, V_{CC} = 5.25V
V_{OH}	Output High Voltage: For D(31:0), PD(31:16)	2.4		V	I_{OH} = –5.0mA, V_{CC} = 4.75V
	Output High Voltage: For $\overline{DS}$, $\overline{SWE}$, $\overline{UWE}$, $\overline{LWE}$, PA1, PA0, $\overline{TA}$	2.4		V	I_{OH} = –16mA, V_{CC} = 4.75V
V_{OL}	Output Low Voltage: For D(31:0), PD(31:16)		0.5	V	I_{OL} = +5.0mA, V_{CC} = 4.75V
	Output Low Voltage: For $\overline{DS}$, $\overline{SWE}$, $\overline{UWE}$, $\overline{LWE}$, PA1, PA0, $\overline{TA}$		0.5	V	I_{OL} = +16mA, V_{CC} = 4.75V
ICC	Maximum Quiescent Supply Current		100	µA	V_{CC} = 5.25V

AC SPECIFICATIONS (V_{CC} = 5.0V ±10%; T_A = –40°C to +85°C; C_L = 50pF; R_L = 500Ω)

	Spec Number	Specification/Characteristic	25MHz		33MHz		40MHz		Unit
			Min	Max	Min	Max	Min	Max	
	1	$\overline{CS}$ Asserted to BCLK Rising (Setup)	5		4		4		ns
	2	BCLK Rising to $\overline{CS}$ Negated (Hold)	2		1		1		ns
	3	A1, A0, SIZ1, SIZ0, R/$\overline{W}$ Valid to BCLK Falling (Setup)	8		8		8		ns
	4	BCLK Rising to A1, A0, SIZ1, SIZ0, R/$\overline{W}$ Invalid (Hold)	0		0		0		ns
	5	D31:0 Valid to BCLK Rising [WRITE] (Setup)	3		2		1		ns
	6	BCLK Rising to D31:0 Invalid [WRITE] (Hold)	0		0		0		ns
	7	BCLK Rising to PA1, PA0 Valid (t_{PLH}, t_{PHL})	2	12.5	2	10	2	9	ns
	8	BCLK Rising to PA1, PA0 Invalid (t_{PLH}, t_{PHL})	2	22	2	21	2	20	ns
	9	PD31:16 Valid to BLCK Falling [READ] (Setup)	2		2		2		ns
	10	BCLK Falling to PD31:16 Invalid [READ] (Hold)	8		6		5.5		ns
	11	$\overline{DSACK1}$, $\overline{DSACK0}$ Valid to BCLK Falling (Setup)	4		2		2		ns
	12	BCLK Falling to $\overline{DSACK1}$, $\overline{DSACK0}$ Invalid (Hold)	8		6		5		ns
	13	BCLK Falling to $\overline{DS}$, $\overline{SWE}$, $\overline{UWE}$, $\overline{LWE}$ (t_{PLH}, t_{PHL})	3	10	3	9	3	9	ns
	14	BCLK Rising to $\overline{TA}$ (t_{PLH}, t_{PHL})	2	13	2	10	2	9	ns
	15	BCLK Rising D31:16 Low Impedance [READ] (t_{PZL}, t_{PZH})	2	16	2	15	2	14	ns
	16	BCLK Rising to D15:0 Low Impedance [READ] (t_{PZL}, t_{PZH})	2	16	2	15	2	14	ns
	17	BCLK Rising to D31:0 Valid [READ] (t_{PLH}, t_{PHL})	0	3	0	2	0	1	ns
	18a	BCLK Rising to D31:16 High Impedance [READ] (t_{PLZ}, t_{PHZ})	4	18	4	16	3	14	ns
	18b	BCLK Rising to D15:0 High Impedance [READ] (t_{PLZ}, t_{PHZ})	4	22	4	20	3	19	ns
	19	BCLK Rising to PD31:16 Valid [WRITE] (t_{PZL}, t_{PZH})	2	23	2	22	2	21	ns
	20	BCLK Rising to PD31:16 [WRITE] (t_{PLH}, t_{PHL})	4	17	4	16	3	15	ns
	21	BCLK Rising to PD31:16 High Impedance [WRITE] (t_{PLZ}, t_{PHZ})	3	19	3	18	3	17	ns
ASYNCHRONOUS OPERATION ONLY	22	$\overline{DS}$ Asserted to $\overline{SWE}$, $\overline{UWE}$, $\overline{LWE}$ Asserted [WRITE] (Skew)		±0.5		±0.5		±0.5	ns
	23	$\overline{DS}$ Negated to $\overline{SWE}$, $\overline{UWE}$, $\overline{LWE}$ Negated [WRITE] (Skew)		±1.0		±1.0		±1.0	ns
	24	$\overline{DSACK1}$, $\overline{DSACK0}$, Invalid to $\overline{DS}$ Negated (t_{PLH})		4		4		4	ns
	25	$\overline{DS}$ Negated Width	38	42	28	32	23	27	ns
	26	$\overline{SWE}$, $\overline{UWE}$, $\overline{LWE}$ Negated Width	38	42	28	32	23	27	ns
	27	$\overline{CS}$ to $\overline{DSACK1}$, $\overline{DSACK0}$ Valid	23		21		20		ns
	28	PD31:16 Valid to $\overline{DS}$, $\overline{SWE}$, $\overline{UWE}$, $\overline{LWE}$ Asserted [WRITE]		23		22		21	ns
	29	$\overline{SWE}$, $\overline{LWE}$, $\overline{UWE}$ Negated to PD31:16 Invalid [WRITE]		27		26		25	ns
	30	$\overline{DS}$ Negated to PD31:16 High Impedance [WRITE]		27		26		25	ns
	31	$\overline{DS}$ Negated to PD31:16 Invalid [READ]		3		2		2	ns
	32	$\overline{DS}$, $\overline{SWE}$, $\overline{UWE}$, $\overline{LWE}$ Asserted Width	38	42	28	32	23	27	ns
	33a	$\overline{CS}$ Negated to D31:16 High Impedance [READ] (t_{PHZ}, t_{PLZ})	15	75	15	55	15	45	ns
	33b	$\overline{CS}$ Negated to D15:0 High Impedance [READ] (t_{PHZ}, t_{PLZ})	20	70	20	50	20	40	ns
	34	PD31:16 Valid to $\overline{DS}$ Negated [READ]	8		8		8		ns
	35	$\overline{CS}$ Negated to BCLK Rising (Setup) Guarantees Sync Termination	3		3		3		ns
	36	BCLK Rising to $\overline{CS}$ Negated (Hold) Guarantees Async Termination	4		4		4		ns
	37a	$\overline{TA}$ Negation to D31:16 High Impedance [READ] (Async Termination)		11		10		9	ns
	37b	$\overline{TA}$ Negation to D15:0 High Impedance [READ] (Async Termination)		13		12		11	ns
	38	$\overline{CS}$ Negated to $\overline{TA}$ Negated (Asynchronous Termination)	2	10	2	9	2	8.5	ns
	39a	$\overline{CS}$ Negated to D31:16 High Impedance [READ] (Async Termination)	2	16	2	15	2	14	ns
	39b	$\overline{CS}$ Negated to D15:0 High Impedance [READ] (Async Termination)	3	19	3	17	3	16	ns

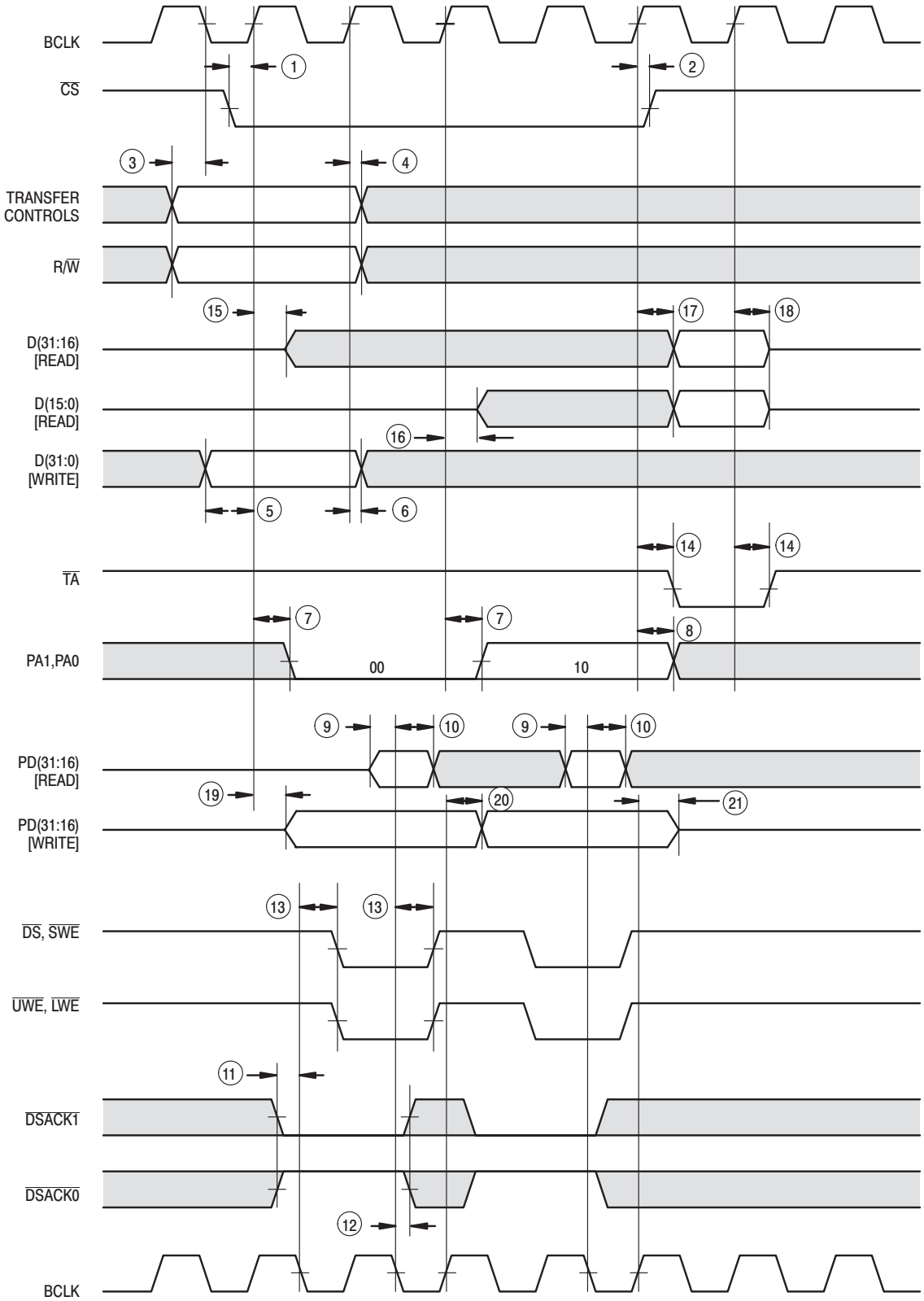


Figure 6. MC68150 READ/WRITE Timing (Two Transfers Shown for Clarification)

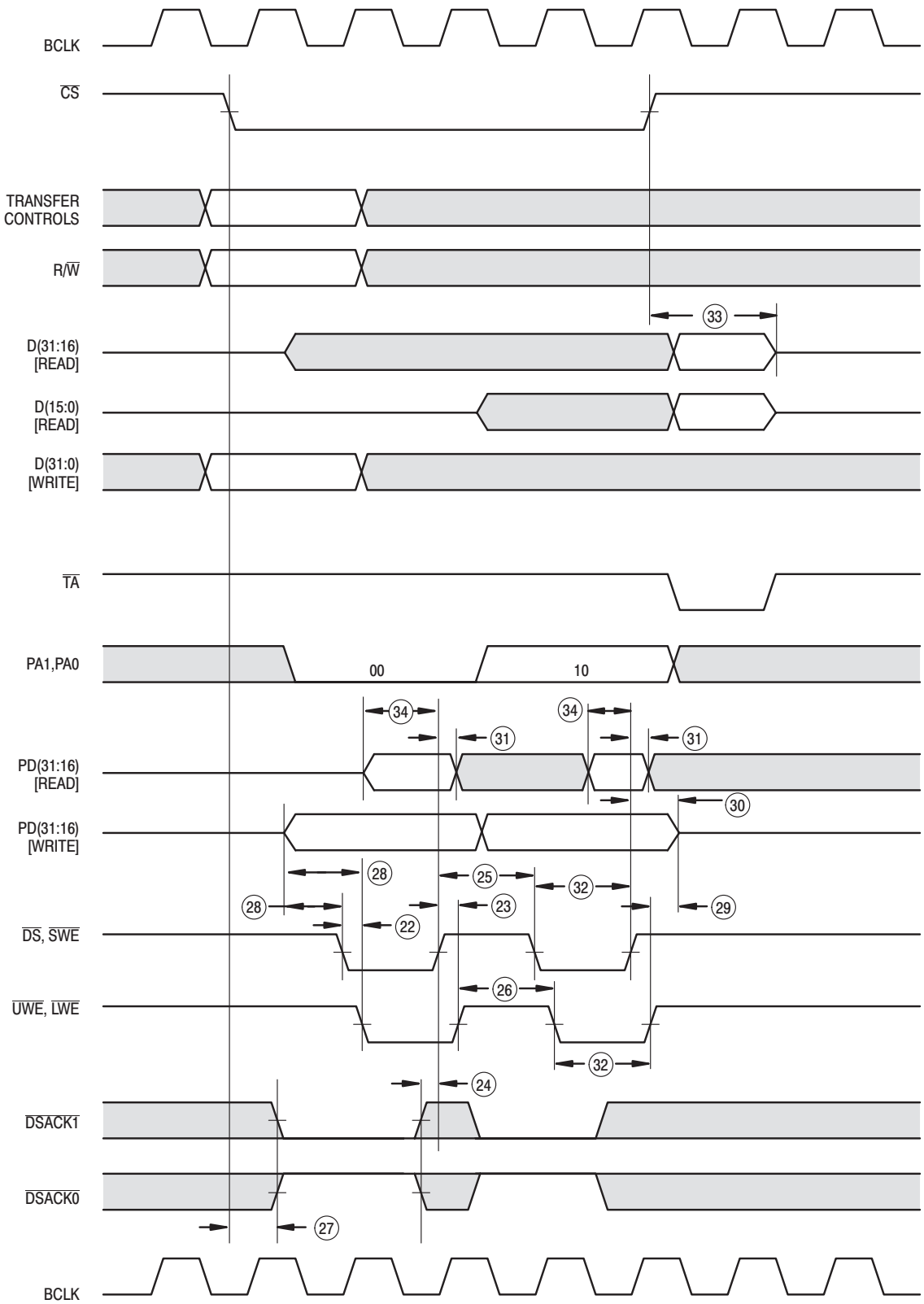


Figure 7. MC68150 READ/WRITE Timing; Asynchronous Operation Only for Processors Other Than 680X0
(Two Transfers Shown for Clarification)

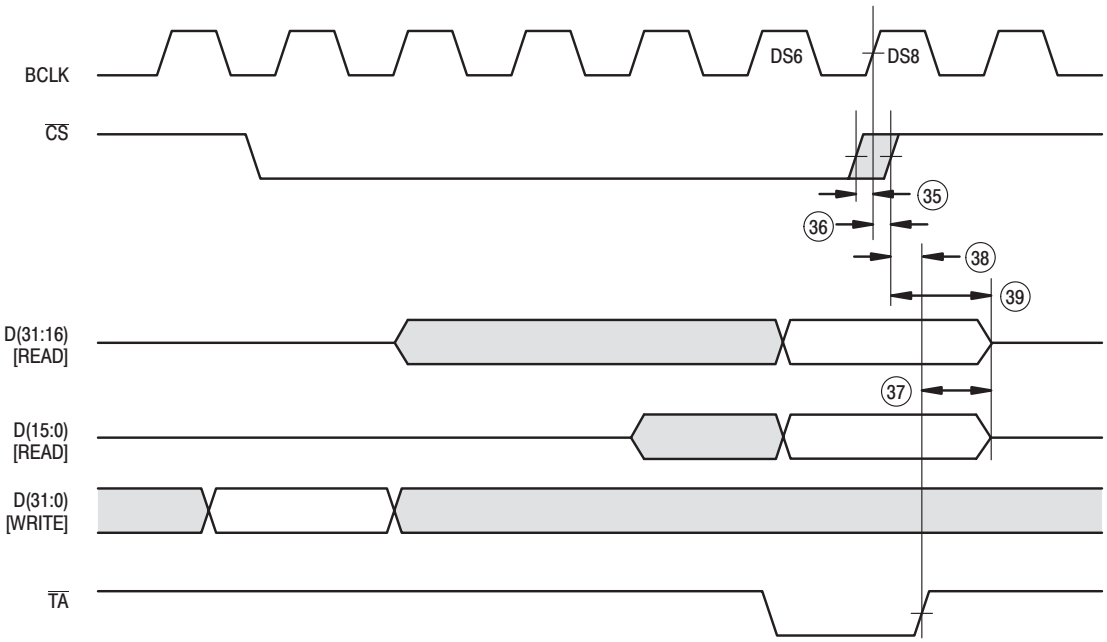


Figure 8. MC68150 READ/WRITE Timing Asynchronous Termination

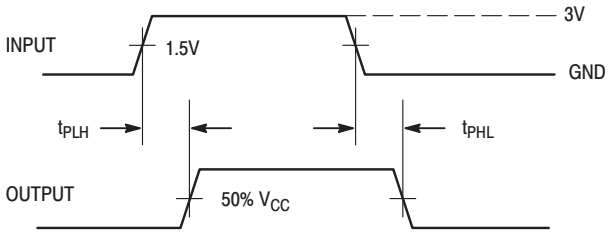


Figure 9. Input/Output Propagation Delays

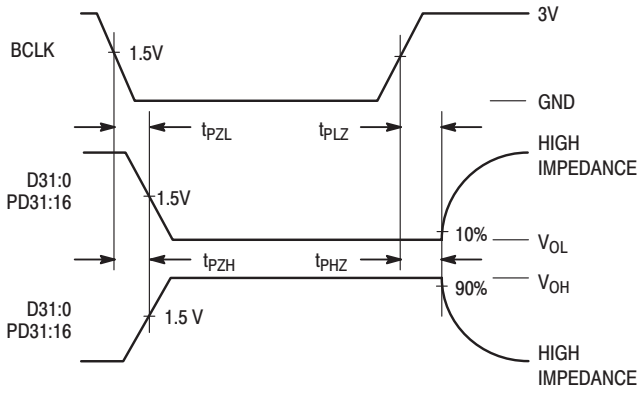


Figure 10. Enable/Disable Times

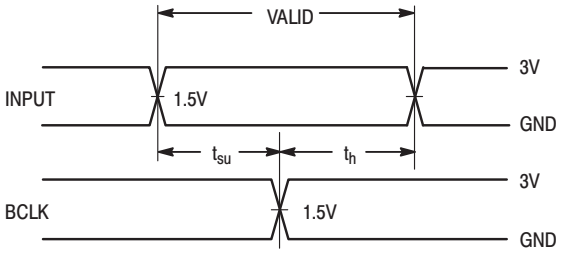


Figure 11. Setup/Hold Times

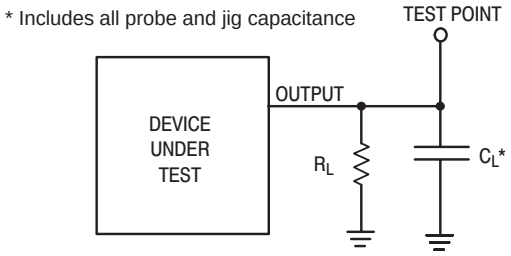


Figure 12. Test Circuit

TIMING DIAGRAM EXAMPLES FOR ALL POSSIBLE TRANSFERS

Figures 13 through 56 depict timing waveforms of all possible transfers, including all address combinations. Each pair of figures includes a read transfer and its corresponding write transfer. Figures 13–20 cover all 32-bit ‘040 bus transfers (to each port size), Figures 21–32 cover all 16-bit ‘040 bus transfers, and Figures 33–56 cover all 8-bit ‘040 bus transfers. Specific timing values have been left off these figures; refer to Figures 6–8 and the AC Specifications for this information.

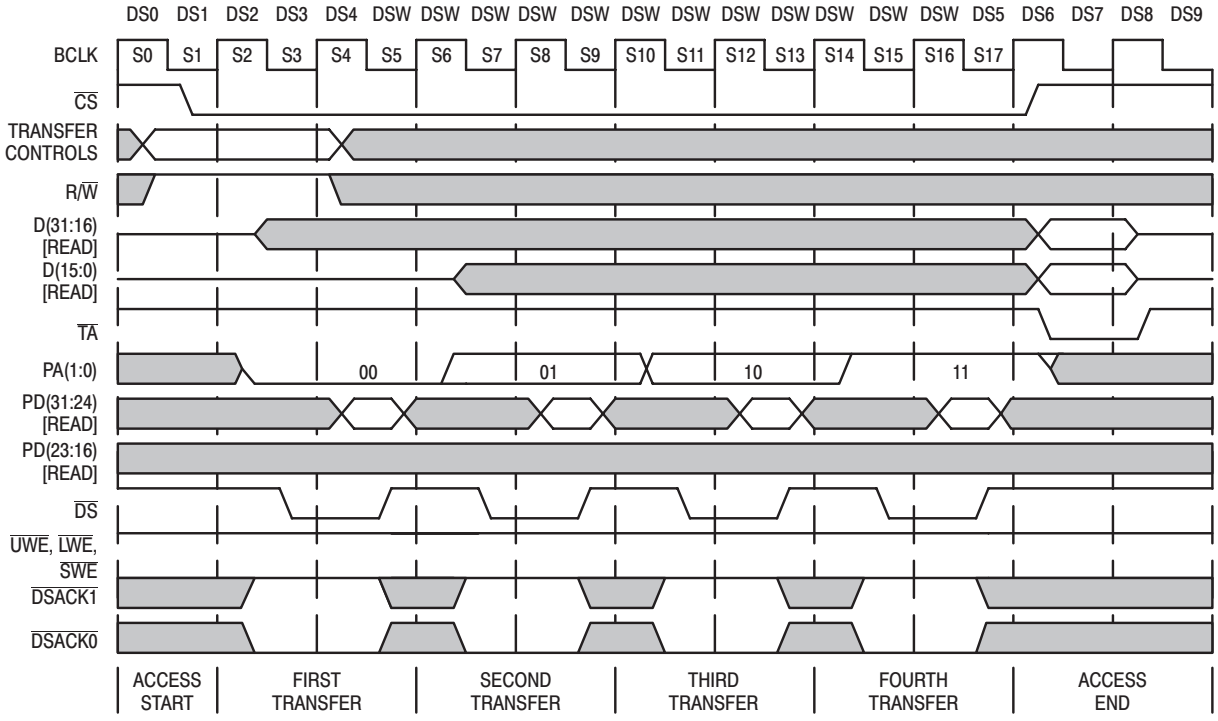


Figure 13. 32-Bit ‘040 READ From 8-Bit Peripheral Example

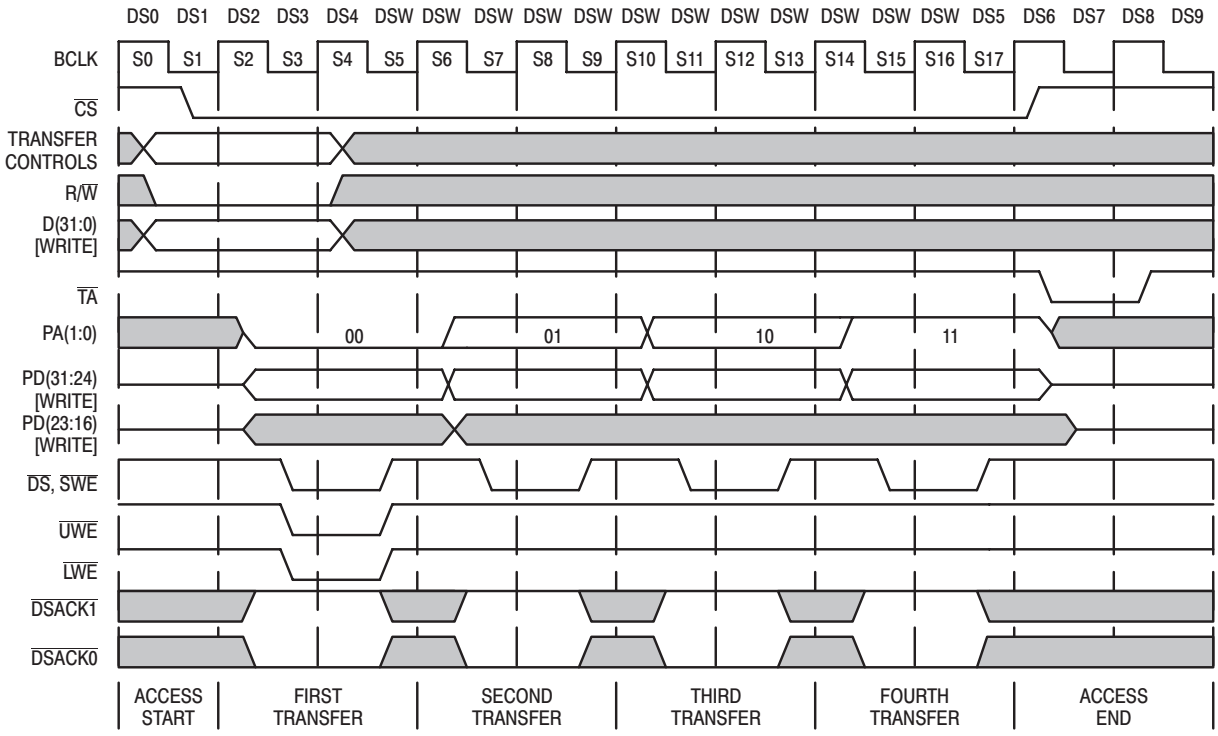


Figure 14. 32-Bit ‘040 WRITE to 8-Bit Peripheral Example

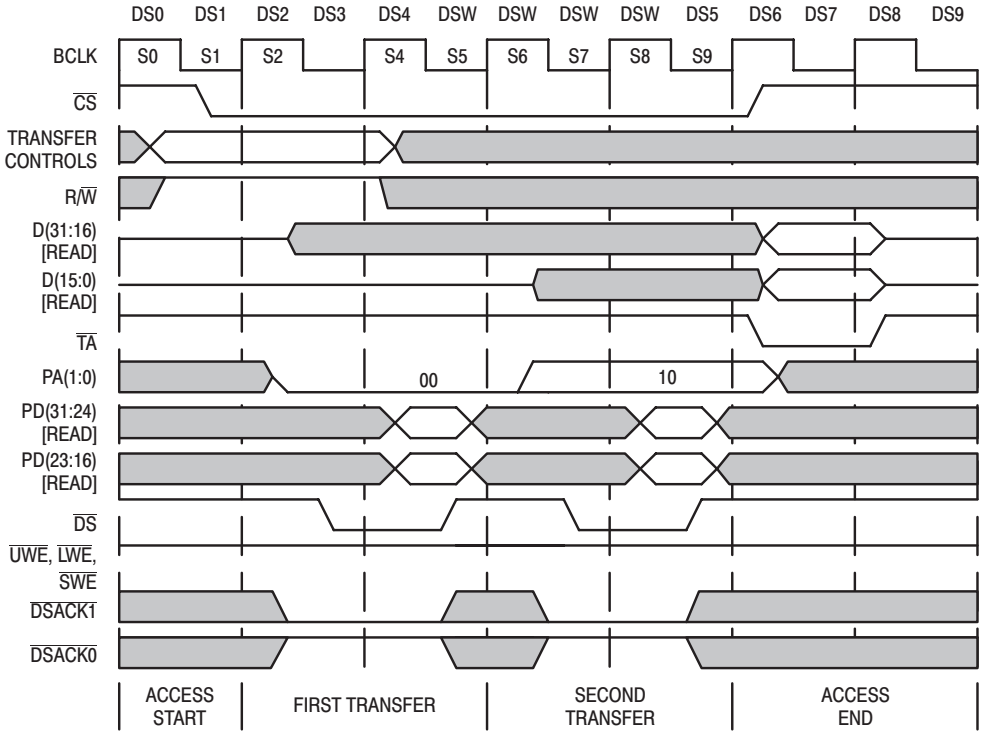


Figure 15. 32-Bit '040 READ From 16-Bit Peripheral Example

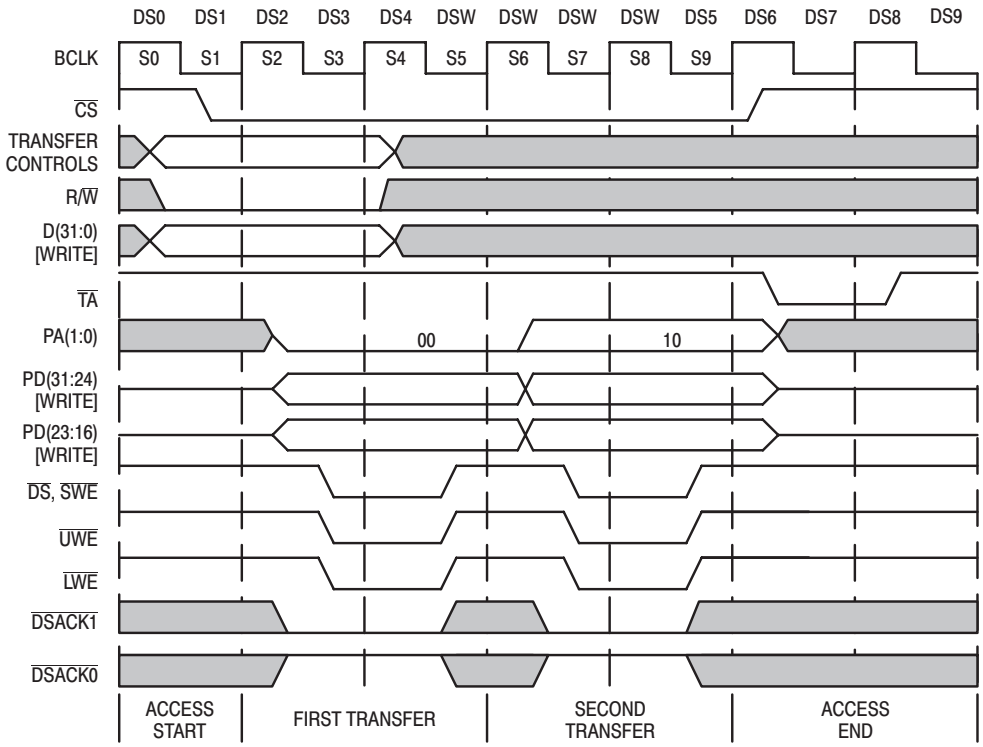


Figure 16.32-Bit '040 WRITE to 16-Bit Peripheral Example

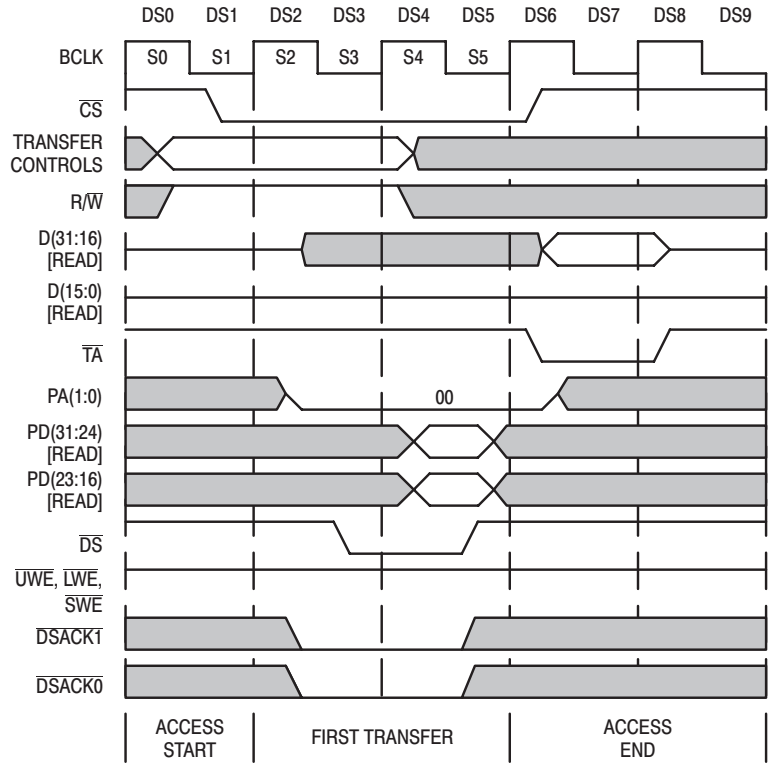


Figure 17. 32-Bit '040 READ From 32-Bit Peripheral Example

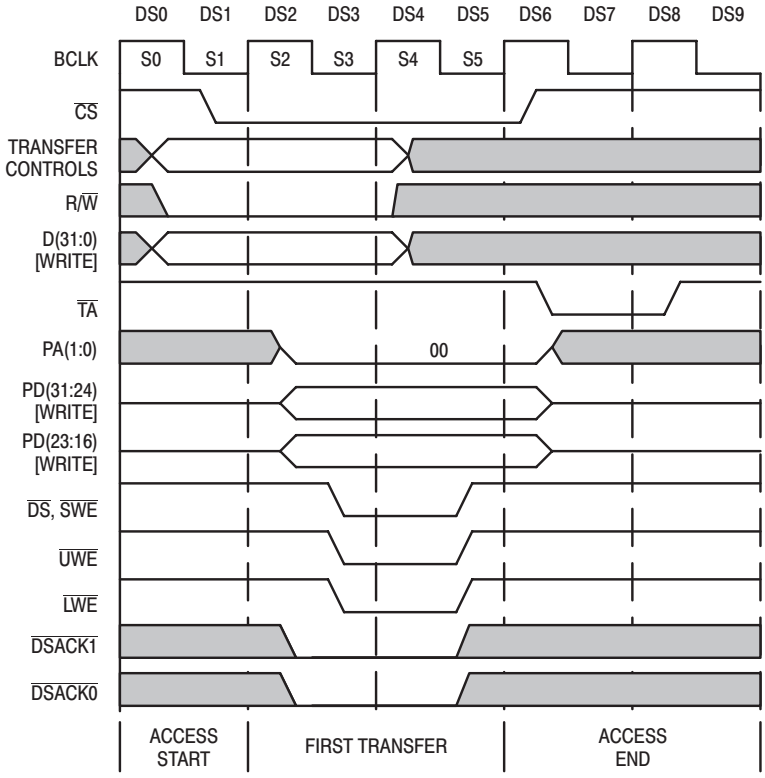


Figure 18. 32-Bit '040 WRITE to 32-Bit Peripheral Example

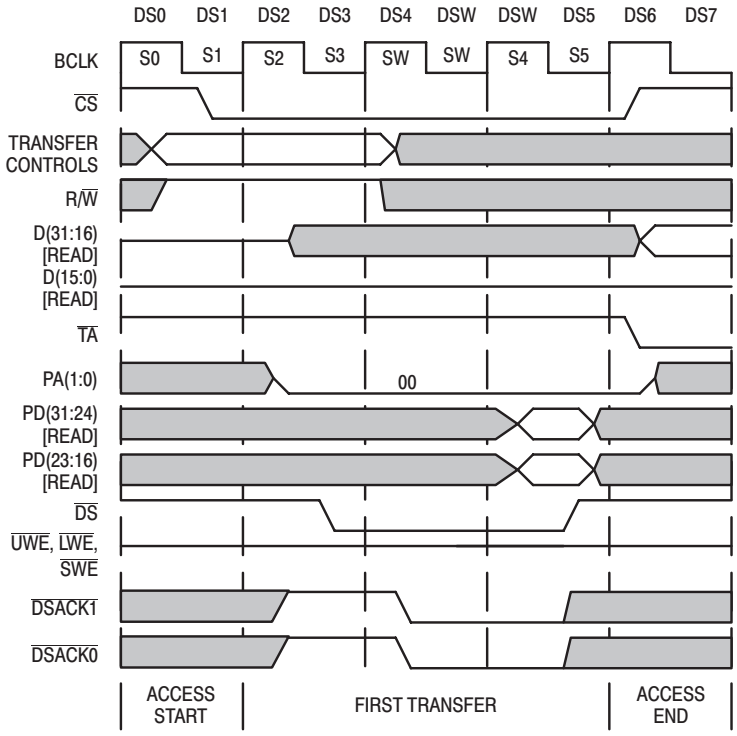


Figure 19. 32-Bit to 32-Bit Transfer With a Wait State on the Peripheral Bus
(This figure purposely shows a partial transfer.)

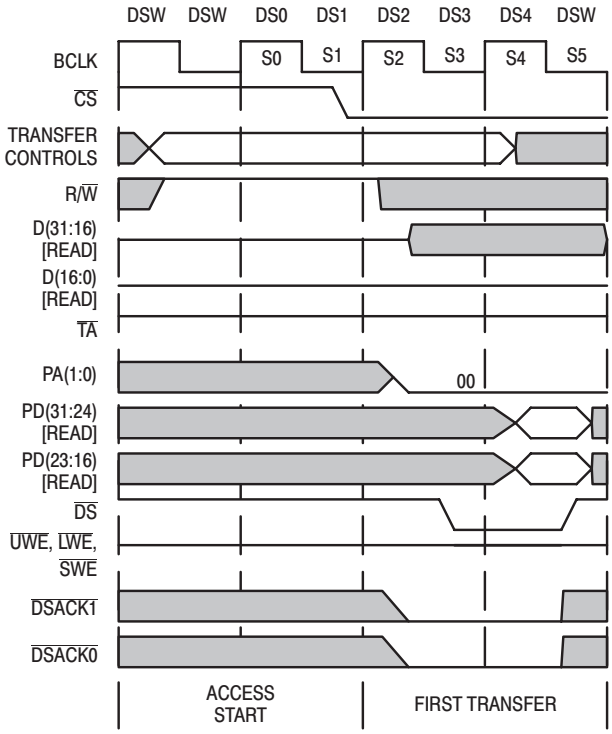


Figure 20. 32-Bit to 32-Bit Transfer With a One Cycle Delayed Start
(This figure purposely shows a partial transfer.)

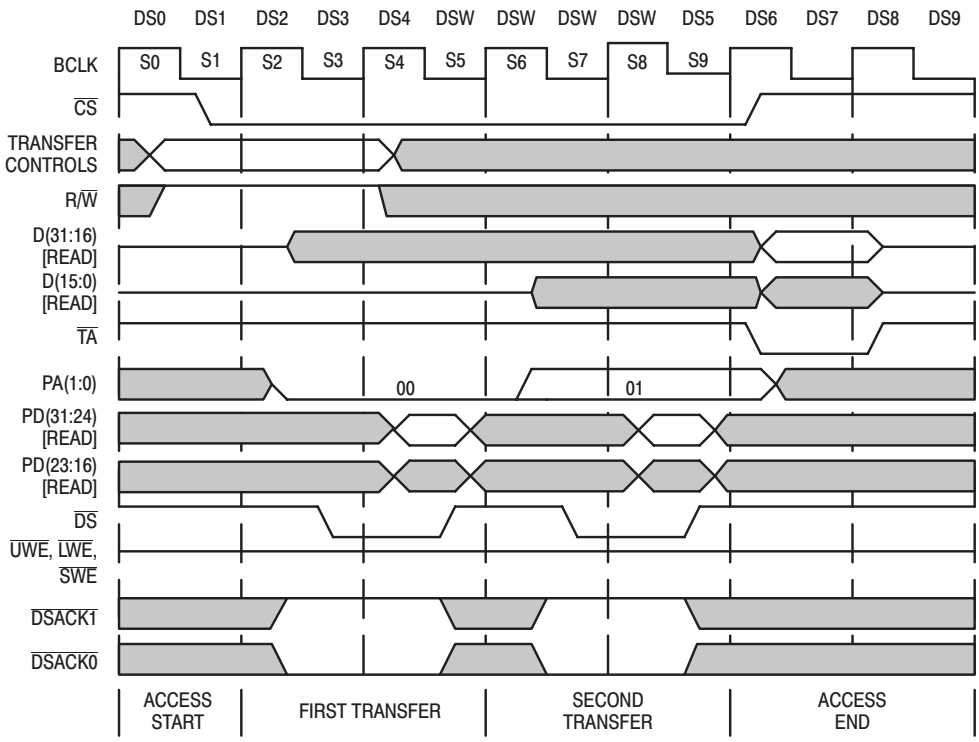


Figure 21. 16-Bit '040 READ From 8-Bit Peripheral Example (PA1=0)

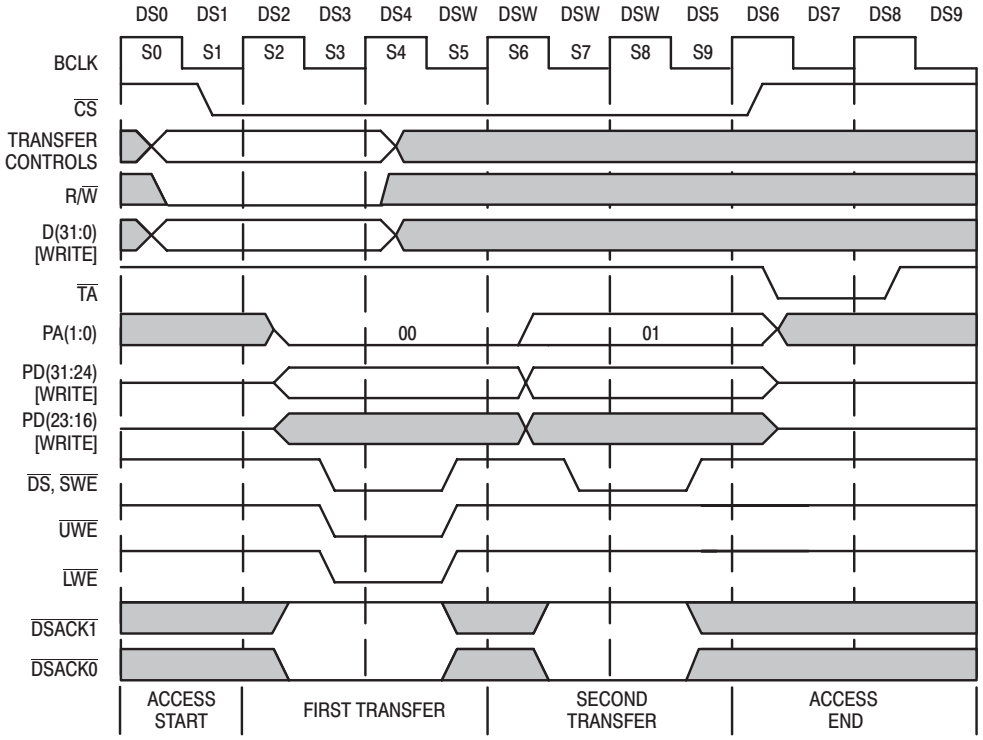


Figure 22. 16-Bit '040 WRITE to 8-Bit Peripheral Example (PA1=0)

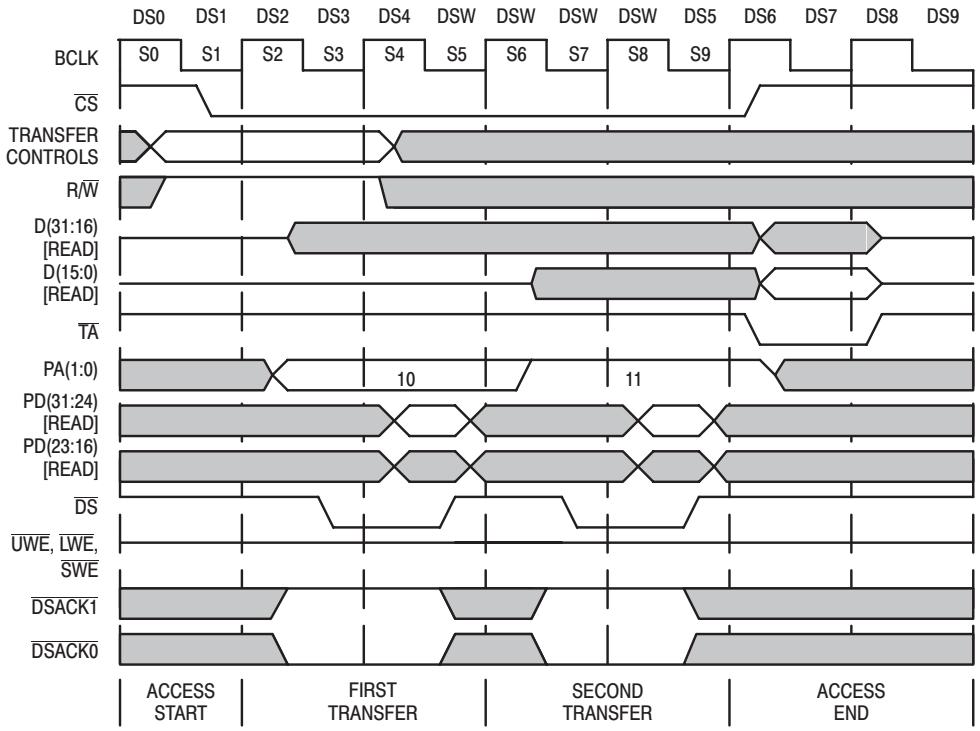


Figure 23. 16-Bit '040 READ From 8-Bit Peripheral Example (PA1=1)

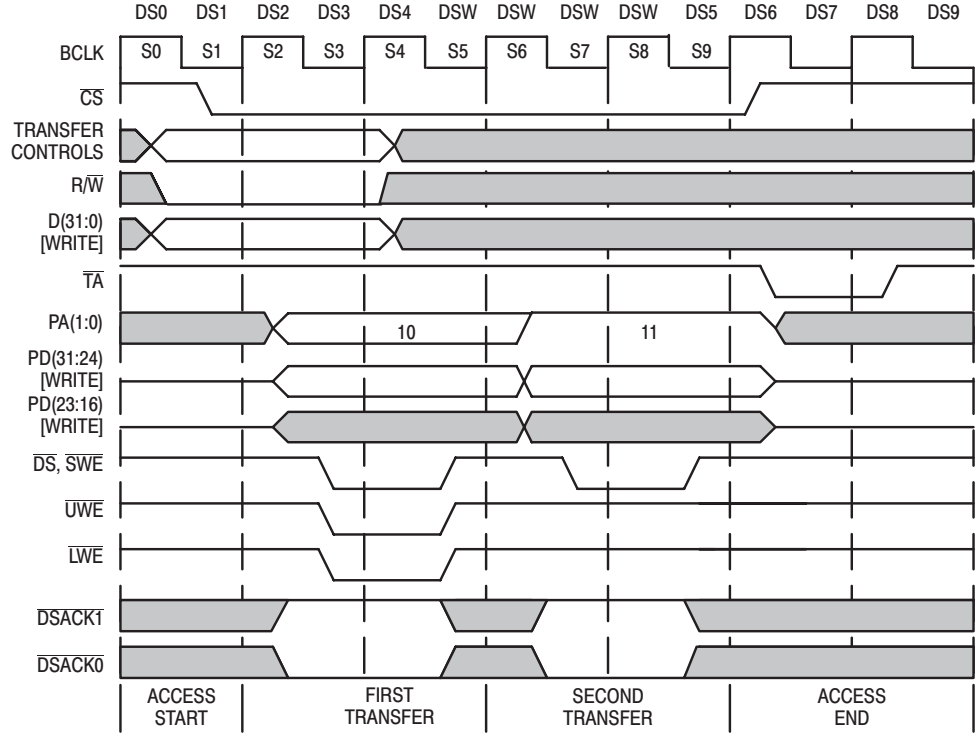


Figure 24. 16-Bit '040 WRITE to 8-Bit Peripheral Example (PA1=1)

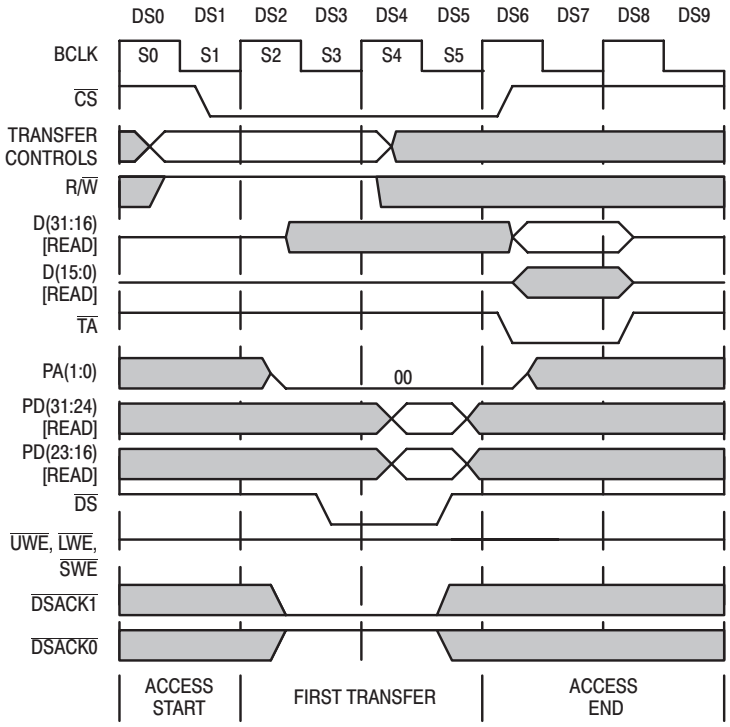


Figure 25. 16 Bit '040 READ From 16-Bit Peripheral Example (PA1=0)

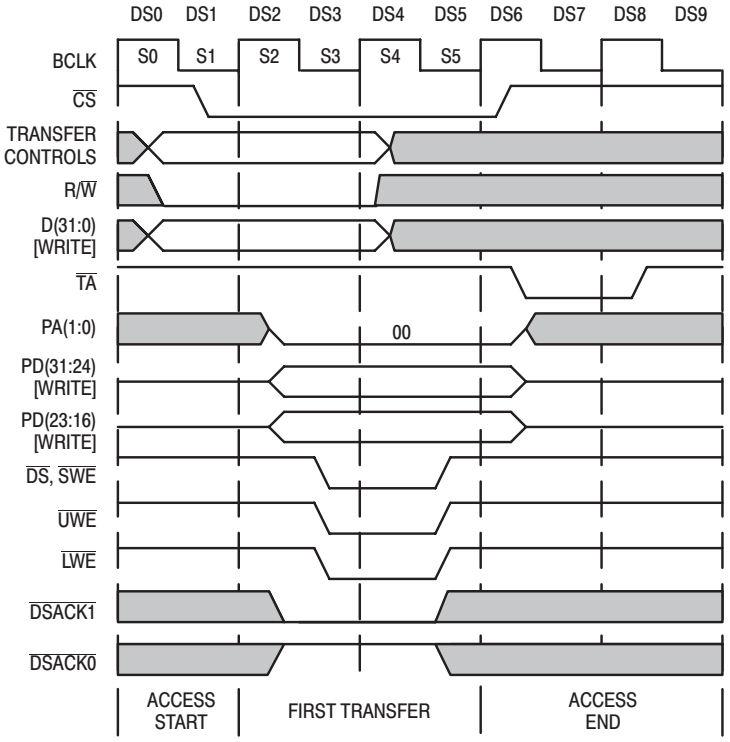


Figure 26. 16-Bit '040 WRITE to 16-Bit Peripheral Example (PA1=0)

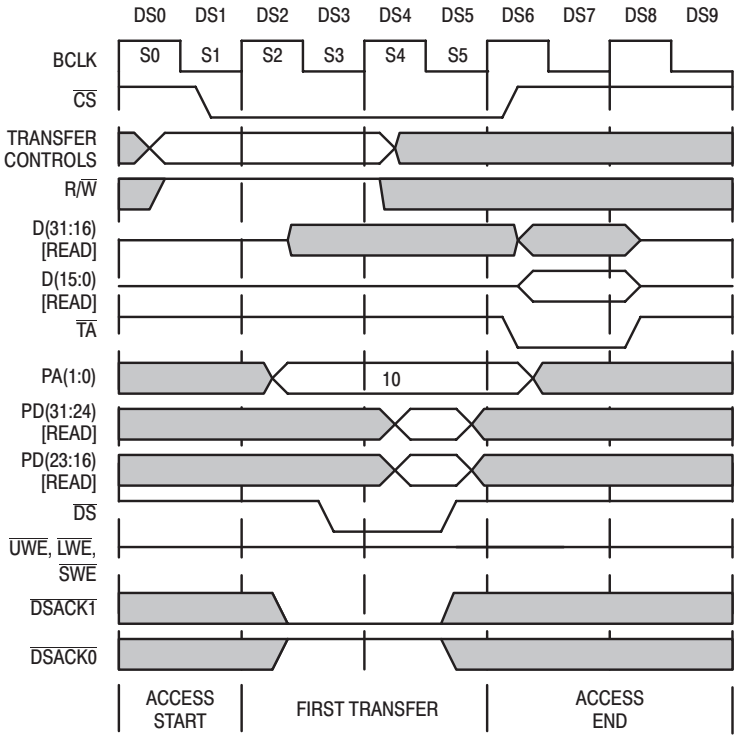


Figure 27. 16-Bit ‘040 READ From 16-Bit Peripheral Example (PA1=1)

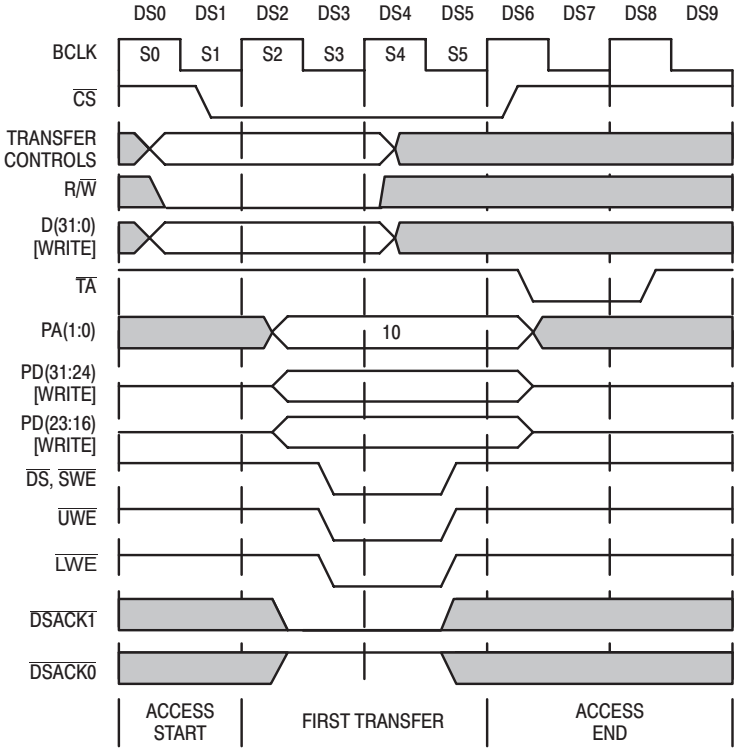


Figure 28. 16-Bit ‘040 WRITE to 16-Bit Peripheral Example (PA1=1)

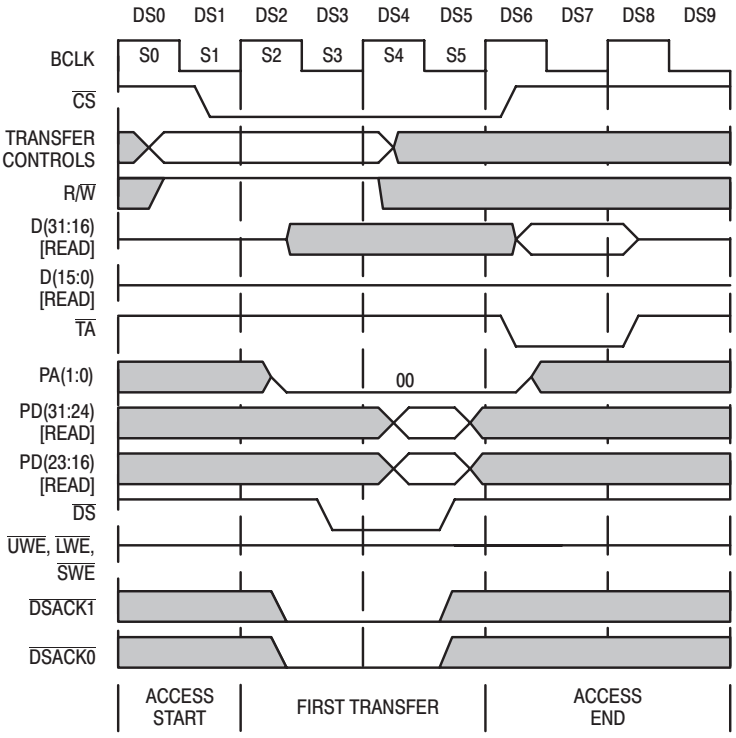


Figure 29. 16-Bit ‘040 READ From 32-Bit Peripheral Example (PA1=0)

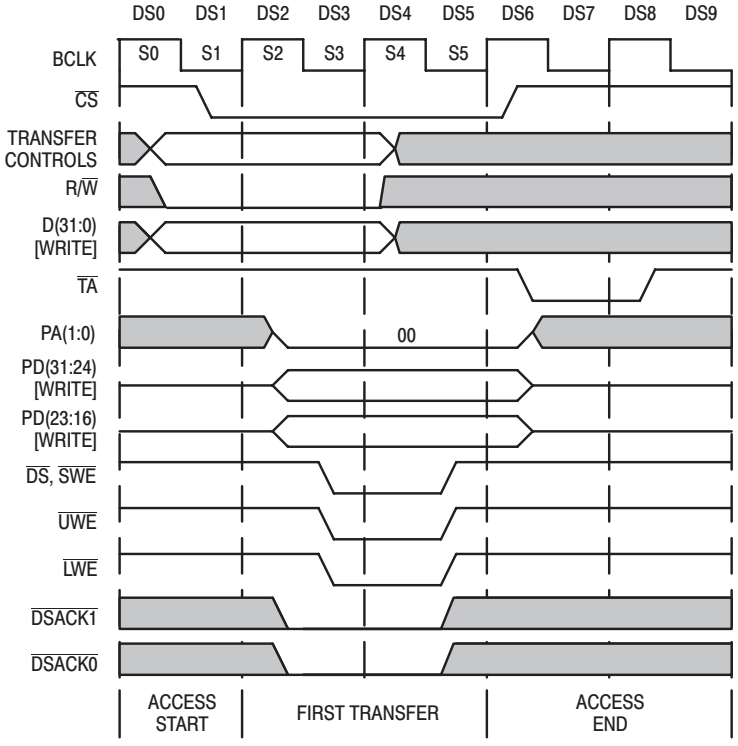


Figure 30. 16-Bit ‘040 WRITE to 32-Bit Peripheral Example (PA1=0)

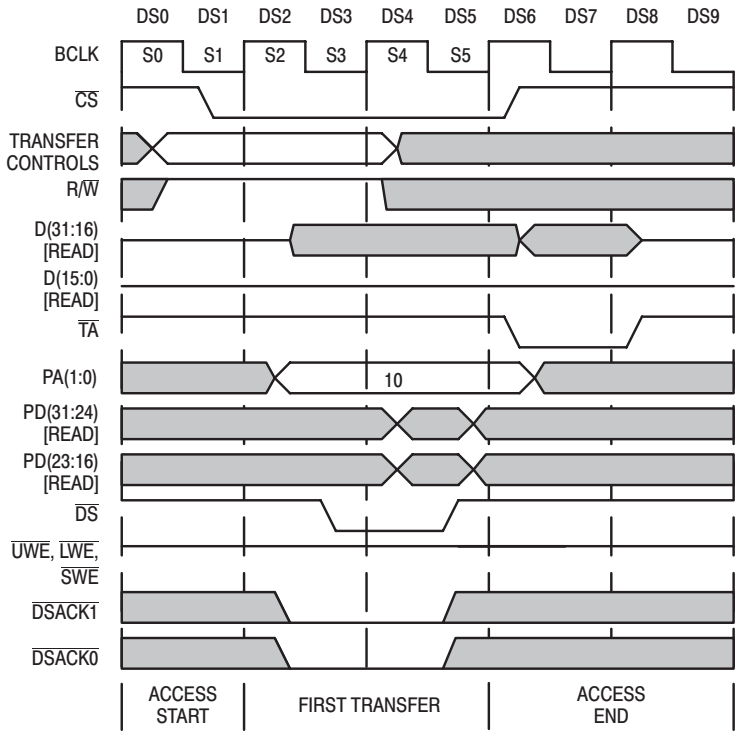


Figure 31. 16-Bit '040 READ From 32-Bit Peripheral Example (PA1=1)

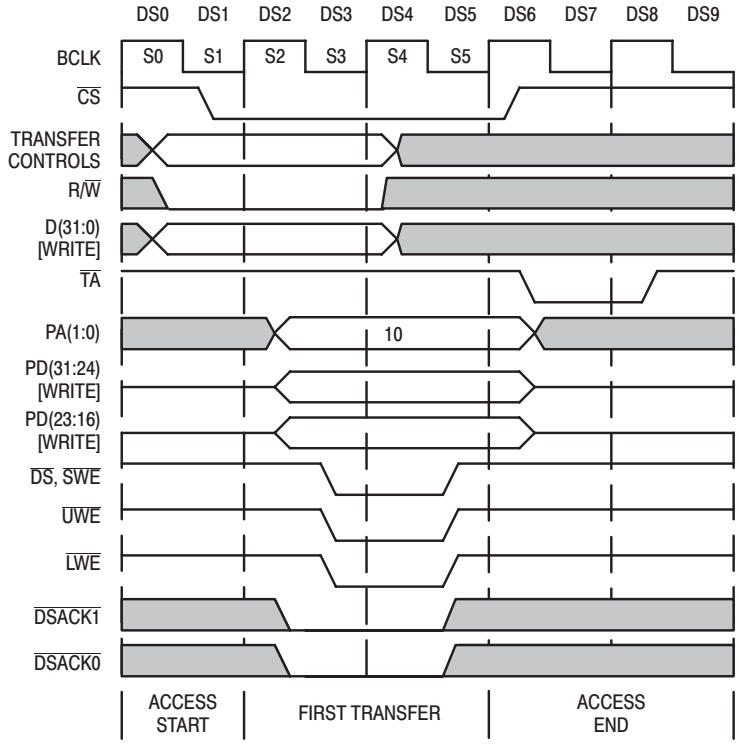


Figure 32. 16-Bit '040 WRITE to 32-Bit Peripheral Example (PA1=1)

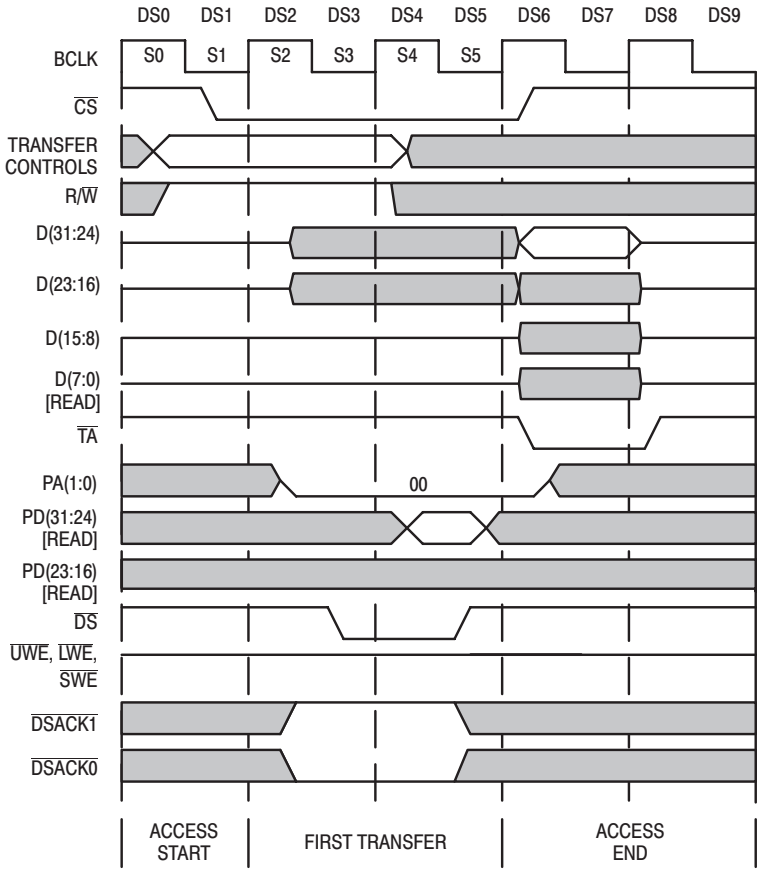


Figure 33. 8-Bit '040 READ From 8-Bit Peripheral Example (PA1,PA0=00)

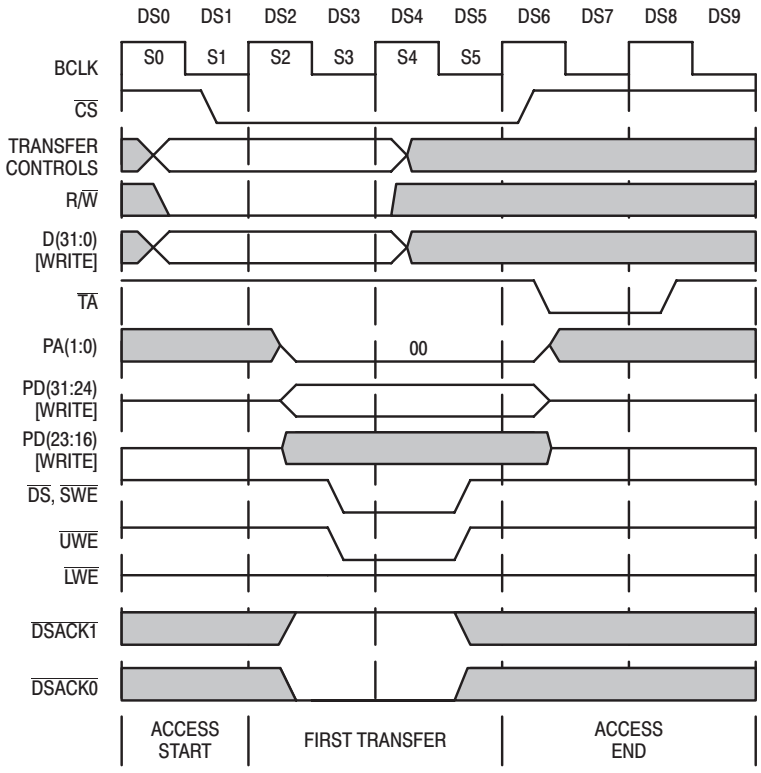


Figure 34. 8-Bit '040 WRITE to 8-Bit Peripheral Example (PA1,PA0=00)

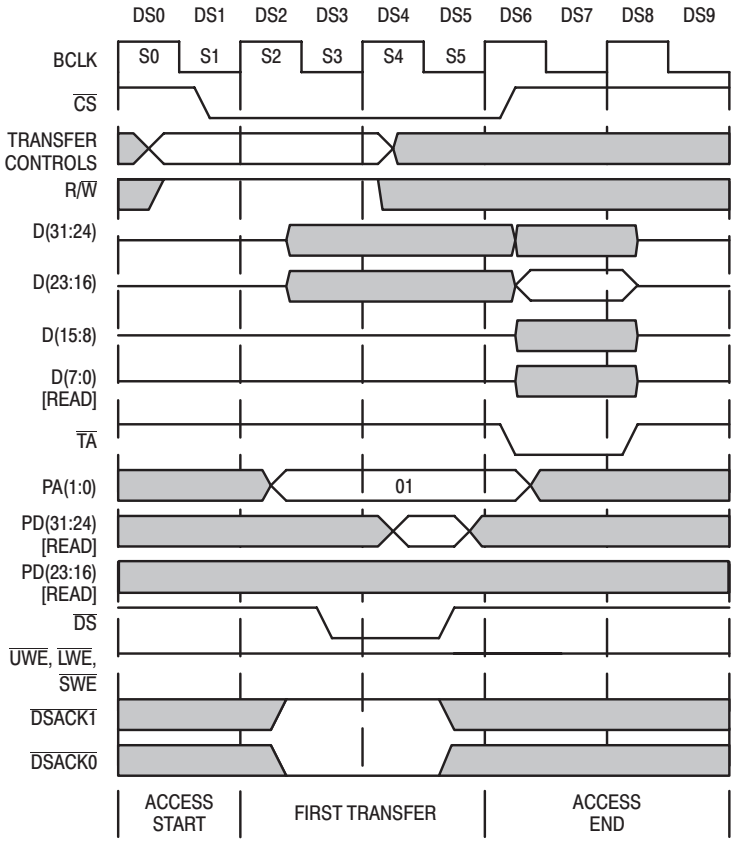


Figure 35. 8-Bit '040 READ From 8-Bit Peripheral Example (PA1,PA0=01)

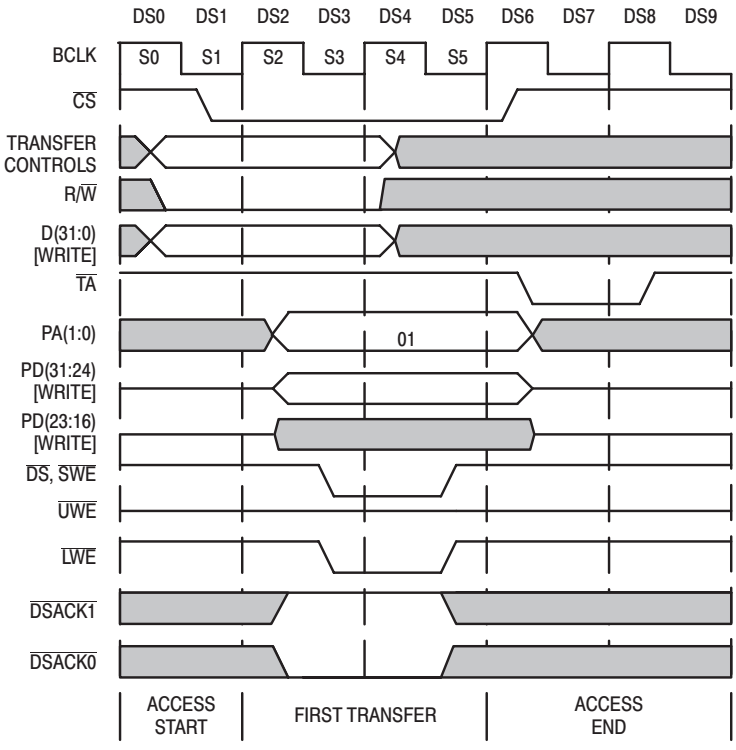


Figure 36. 8-Bit '040 WRITE to 8-Bit Peripheral Example (PA1,PA0=01)

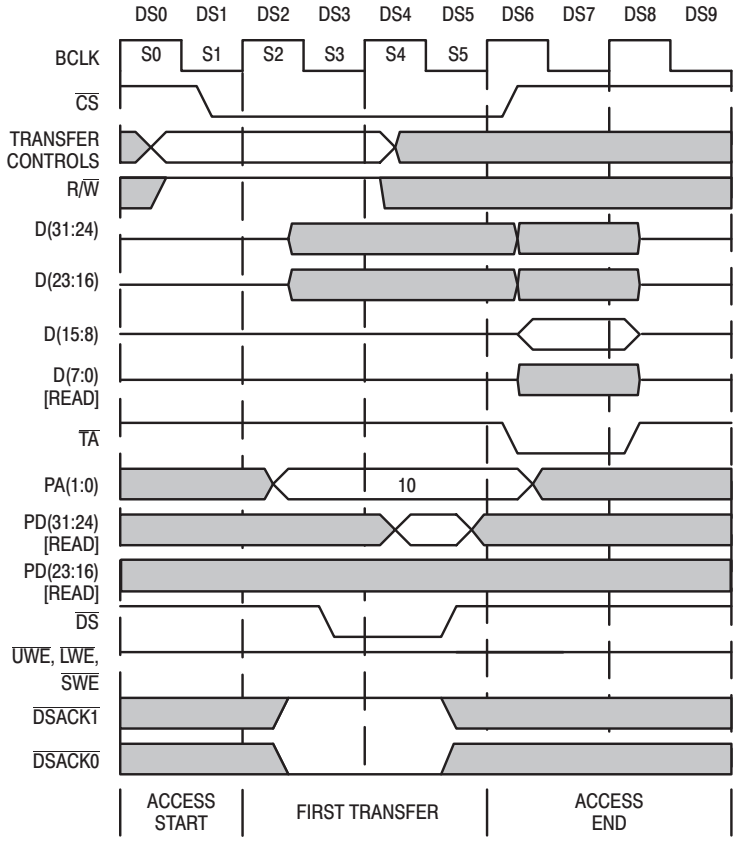


Figure 37. 8-Bit '040 READ From 8-Bit Peripheral Example (PA1,PA0=10)

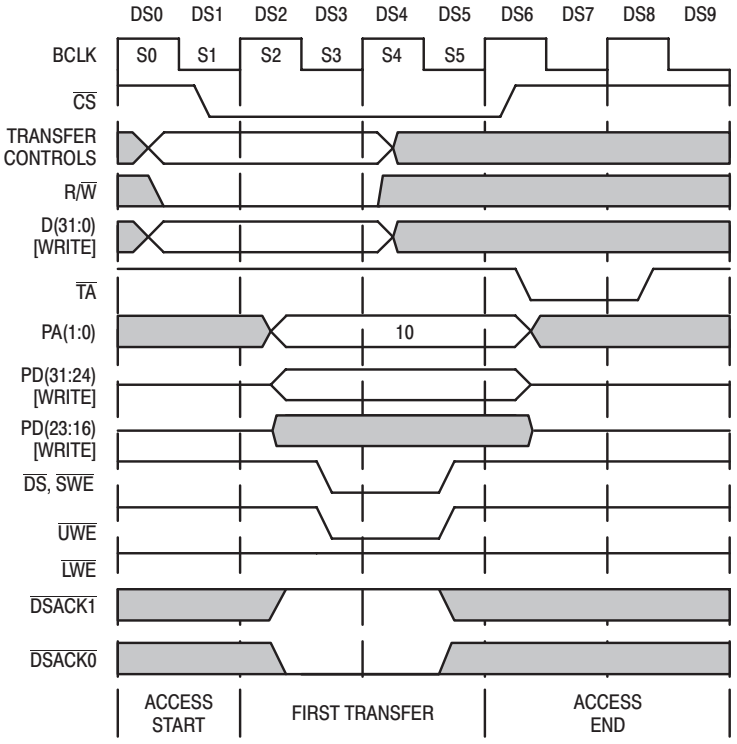


Figure 38. 8-Bit '040 WRITE to 8-Bit Peripheral Example (PA1,PA0=10)

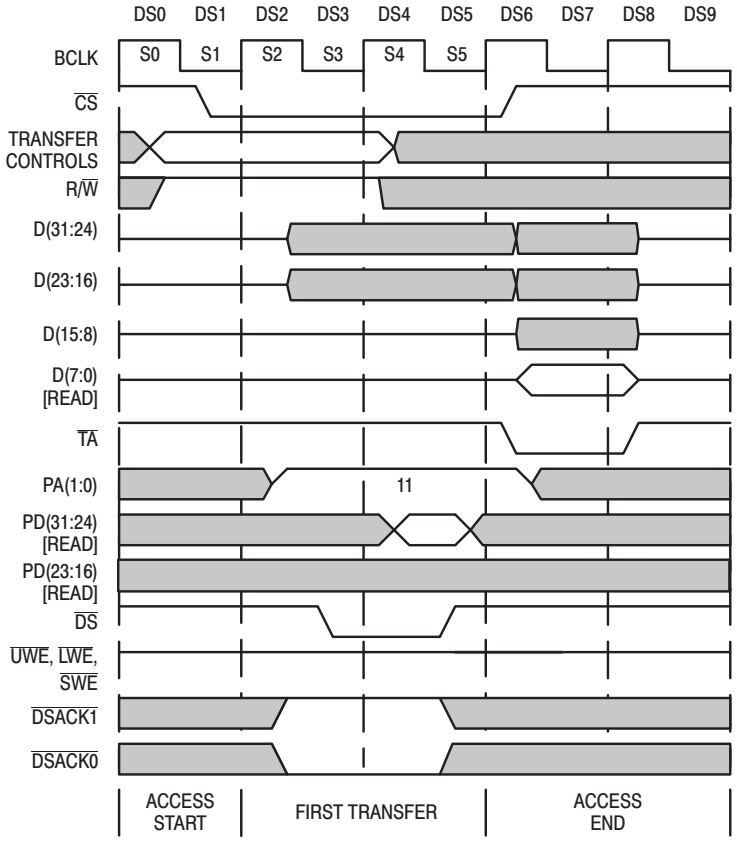


Figure 39. 8-Bit '040 READ From 8-Bit Peripheral Example (PA1,PA0=11)

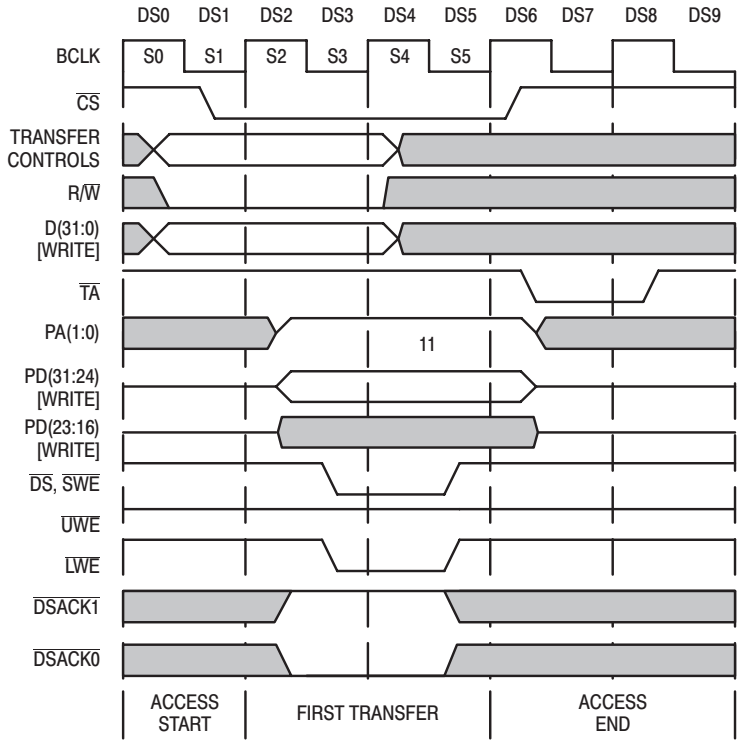


Figure 40. 8-Bit '040 WRITE to 8-Bit Peripheral Example (PA1,PA0=11)

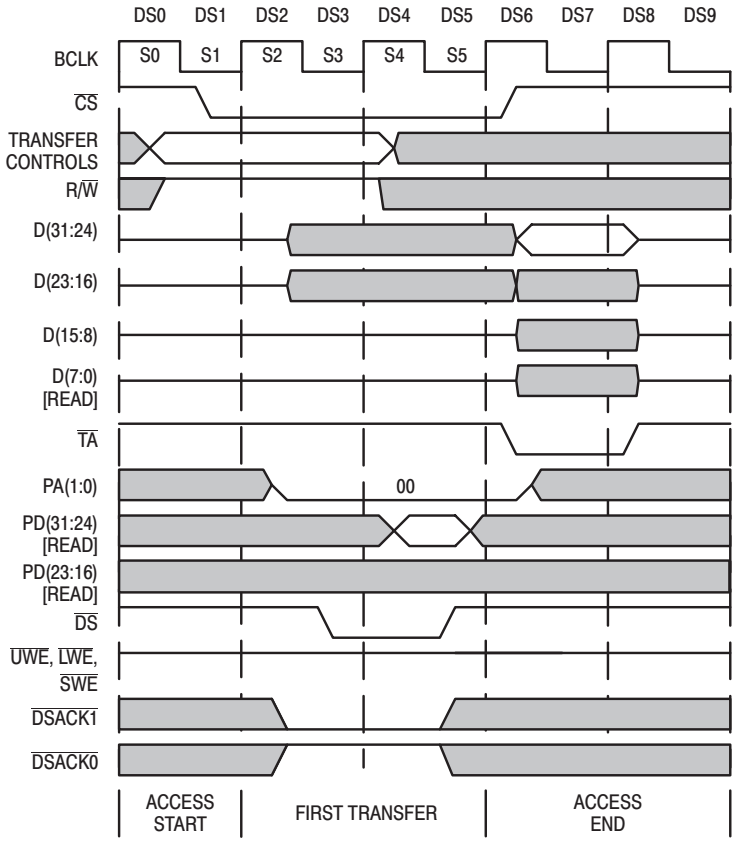


Figure 41. 8-Bit '040 READ From 16-Bit Peripheral Example (PA1,PA0=00)

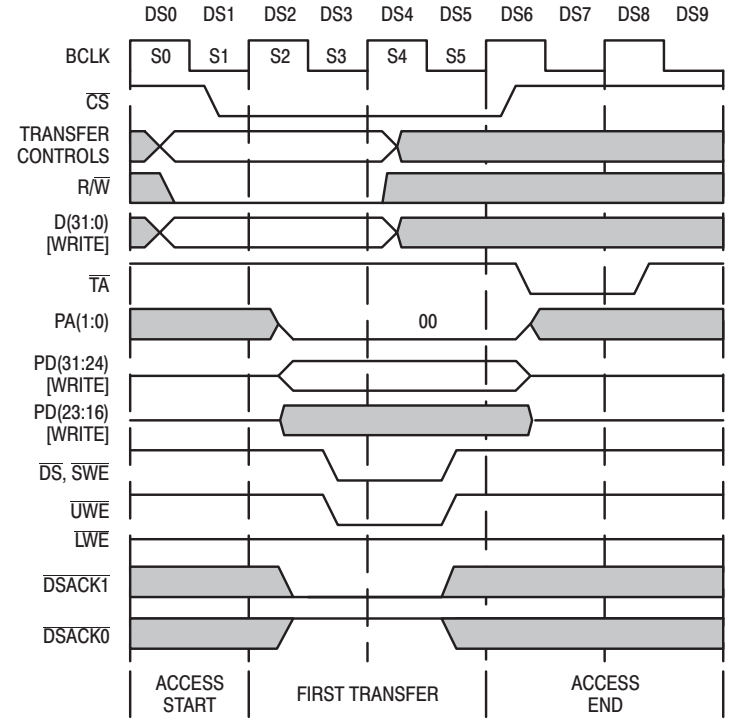


Figure 42. 8-Bit '040 WRITE to 16-Bit Peripheral Example (PA1,PA0=00)

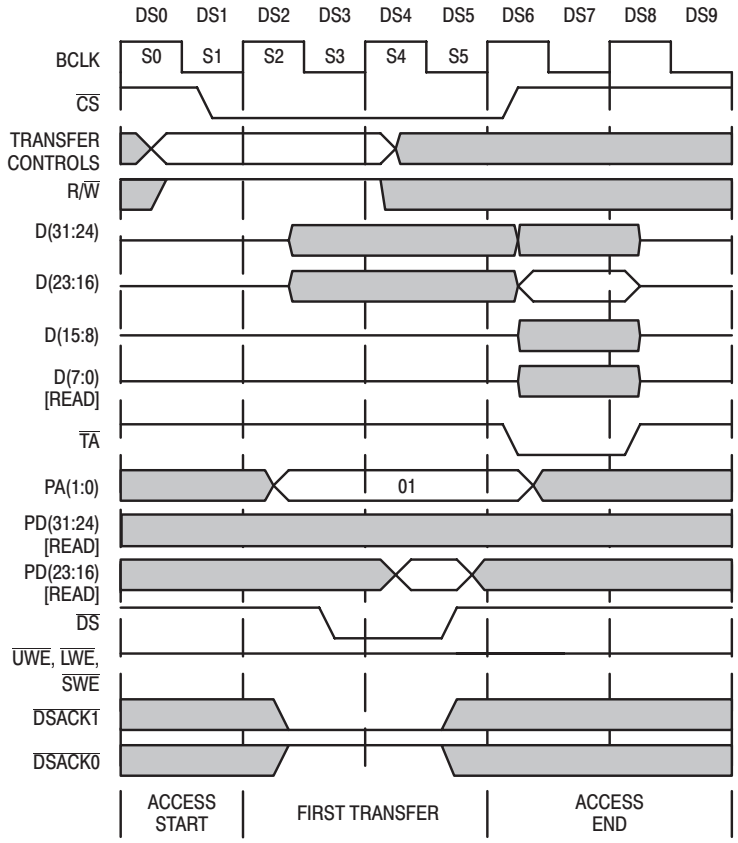


Figure 43. 8-Bit '040 READ From 16-Bit Peripheral Example (PA1,PA0=01)

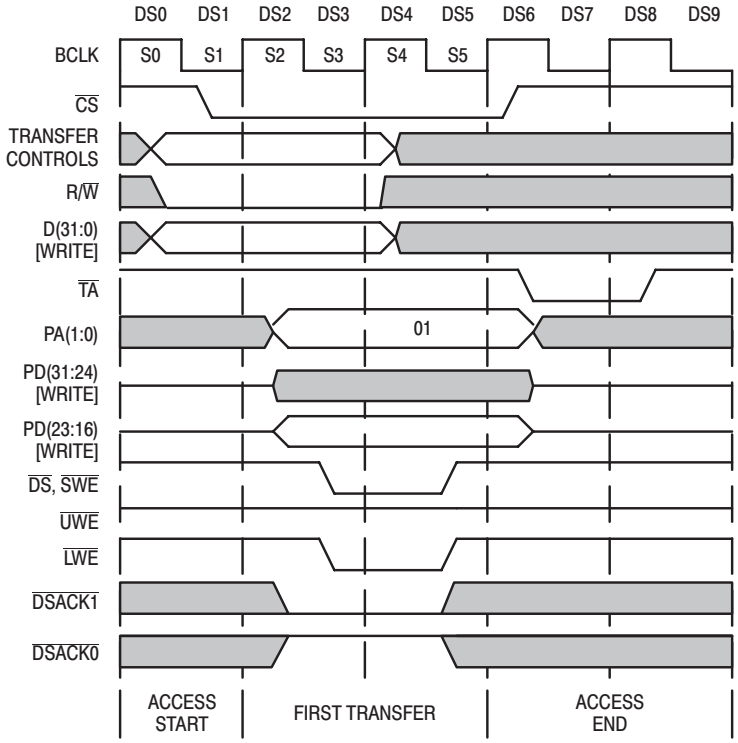


Figure 44. 8-Bit '040 WRITE to 16-Bit Peripheral Example (PA1,PA0=01)

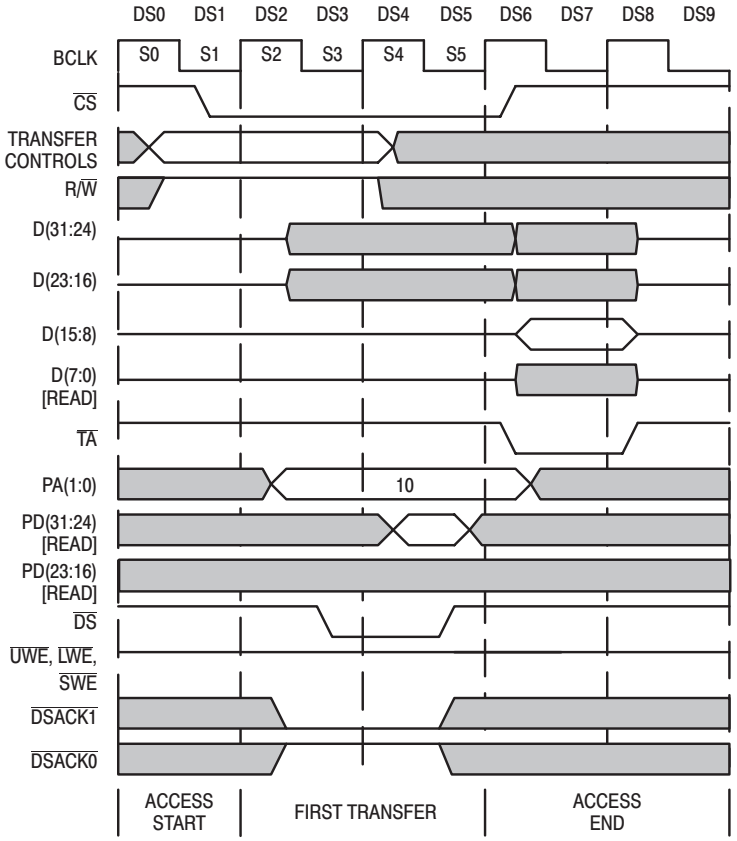


Figure 45. 8-Bit '040 READ From 16-Bit Peripheral Example (PA1,PA0=10)

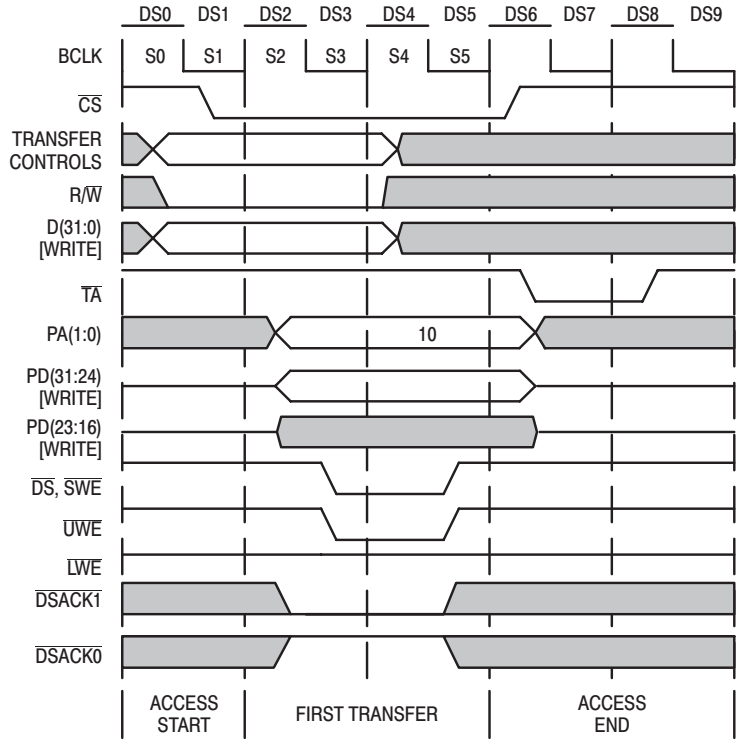


Figure 46. 8-Bit '040 WRITE to 16-Bit Peripheral Example (PA1,PA0=10)

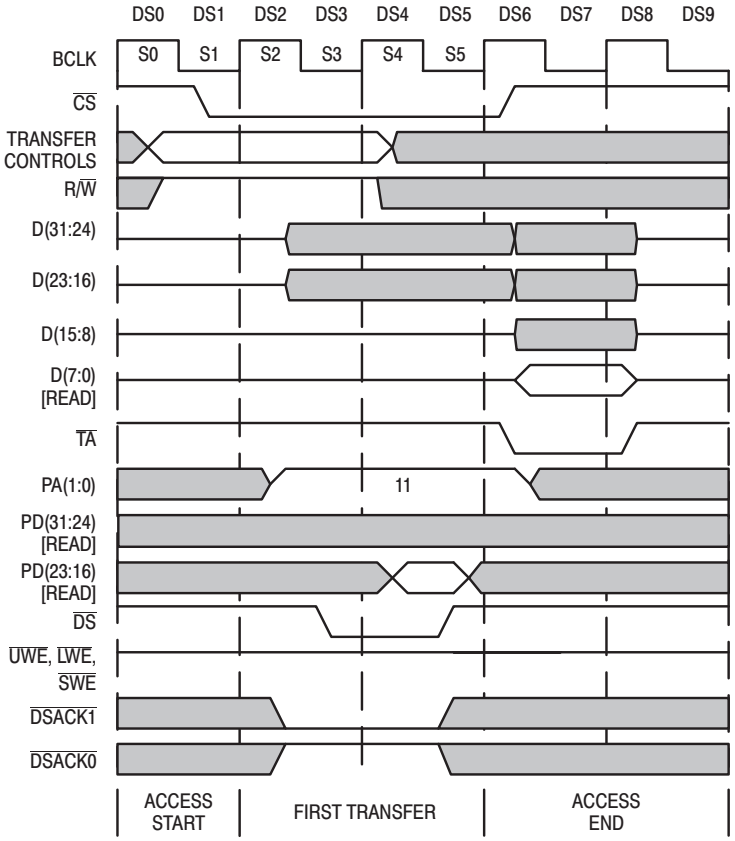


Figure 47. 8-Bit '040 READ From 16-Bit Peripheral Example (PA1,PA0=11)

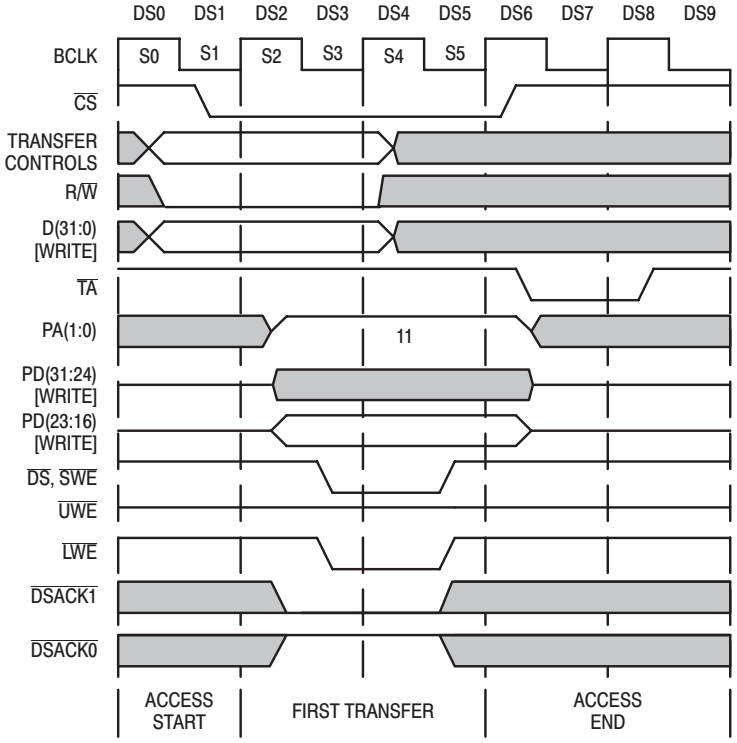


Figure 48. 8-Bit '040 WRITE to 16-Bit Peripheral Example (PA1,PA0=11)

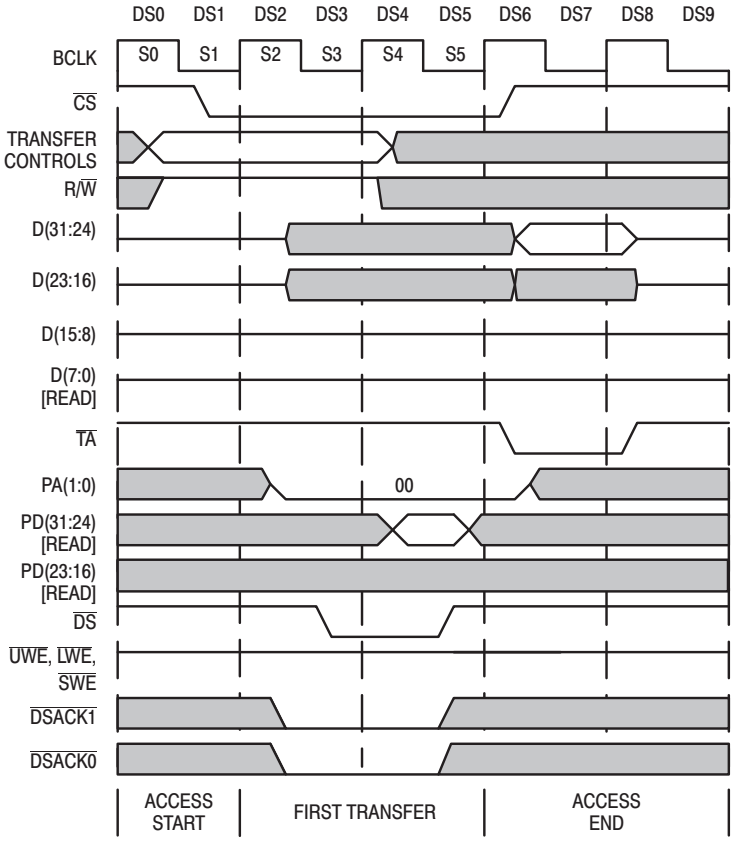


Figure 49. 8-Bit '040 READ From 32-Bit Peripheral Example (PA1,PA0=00)

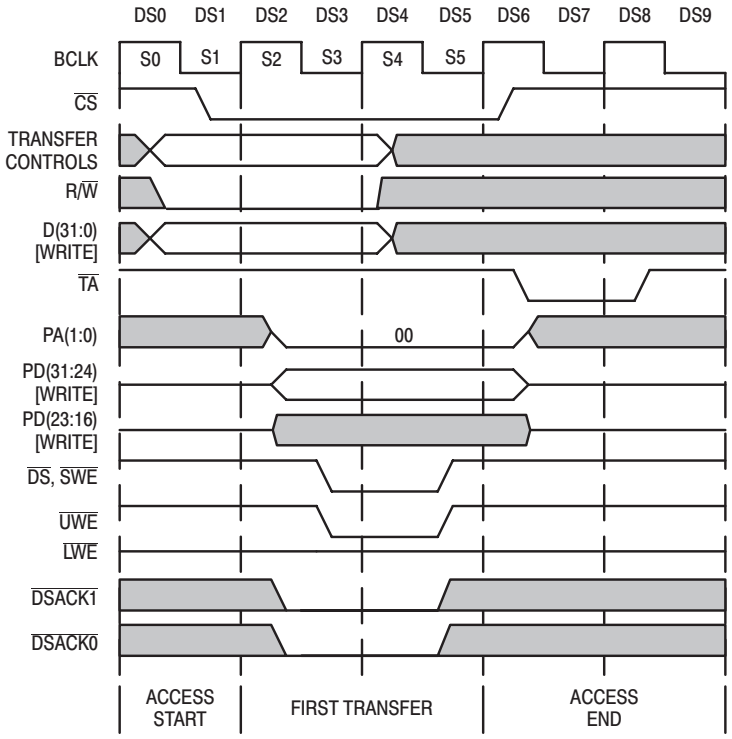


Figure 50. 8-Bit '040 WRITE to 32-Bit Peripheral Example (PA1,PA0=00)

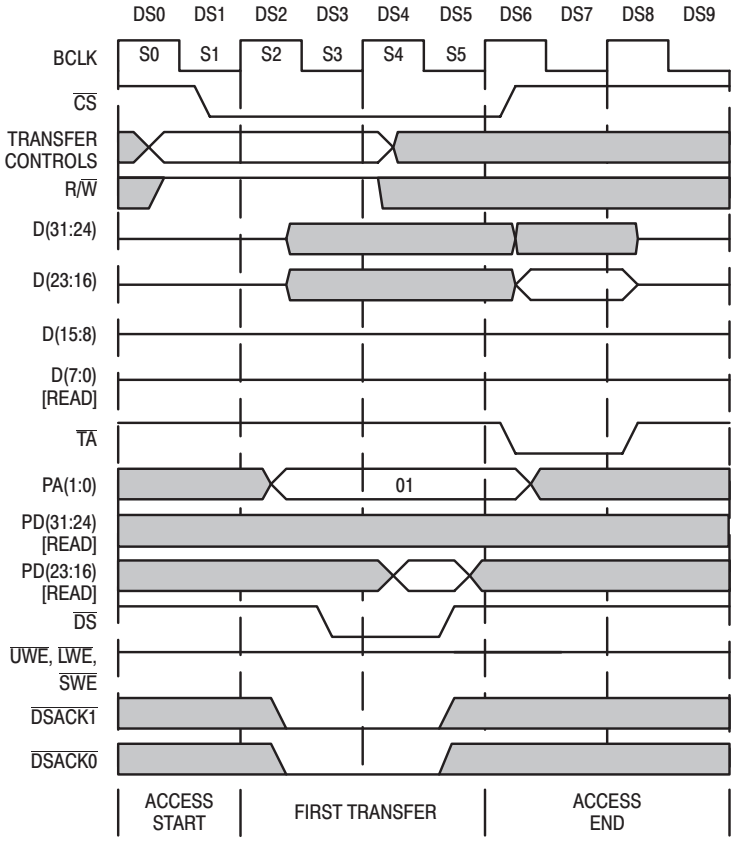


Figure 51. 8-Bit '040 READ From 32-Bit Peripheral Example (PA1,PA0=01)

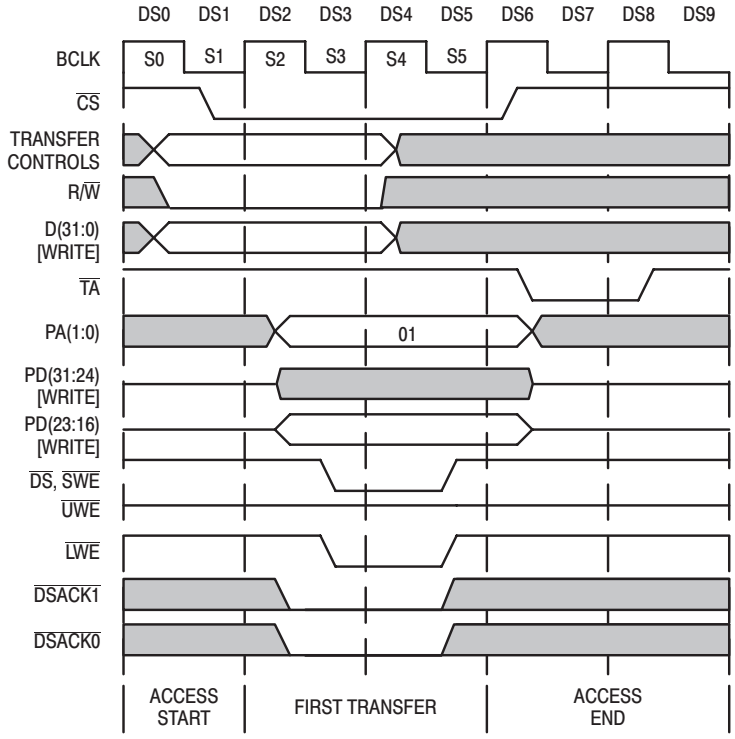


Figure 52. 8-Bit '040 WRITE to 32-Bit Peripheral Example (PA1,PA0=01)

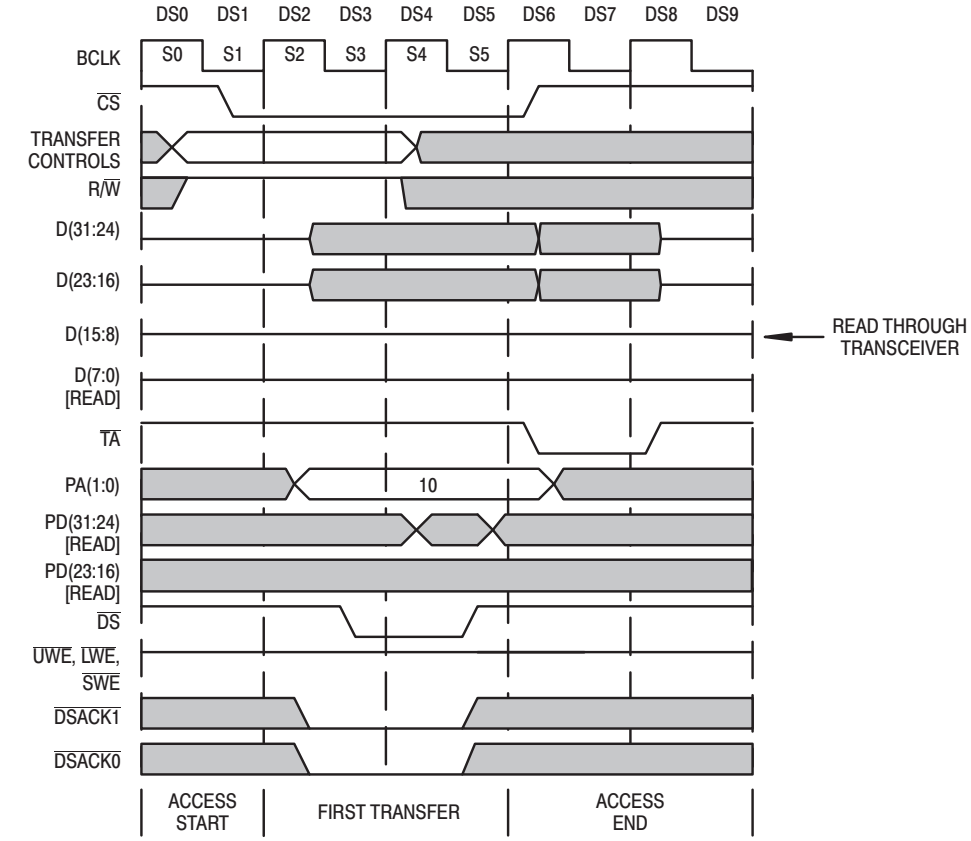


Figure 53. 8-Bit '040 READ From 32-Bit Peripheral Example (PA1,PA0=10)

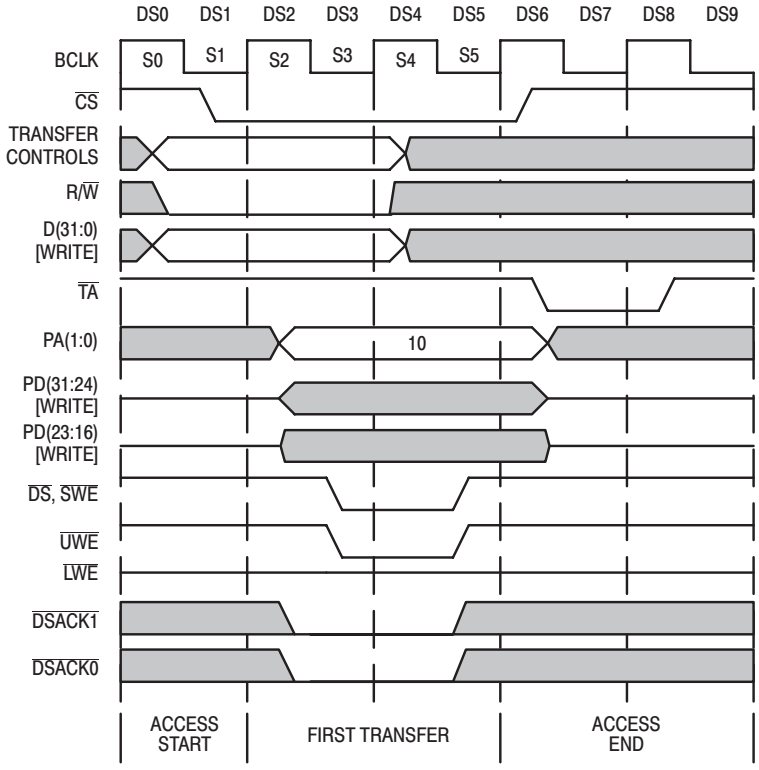


Figure 54. 8-Bit '040 WRITE to 32-Bit Peripheral Example (PA1,PA0=10)

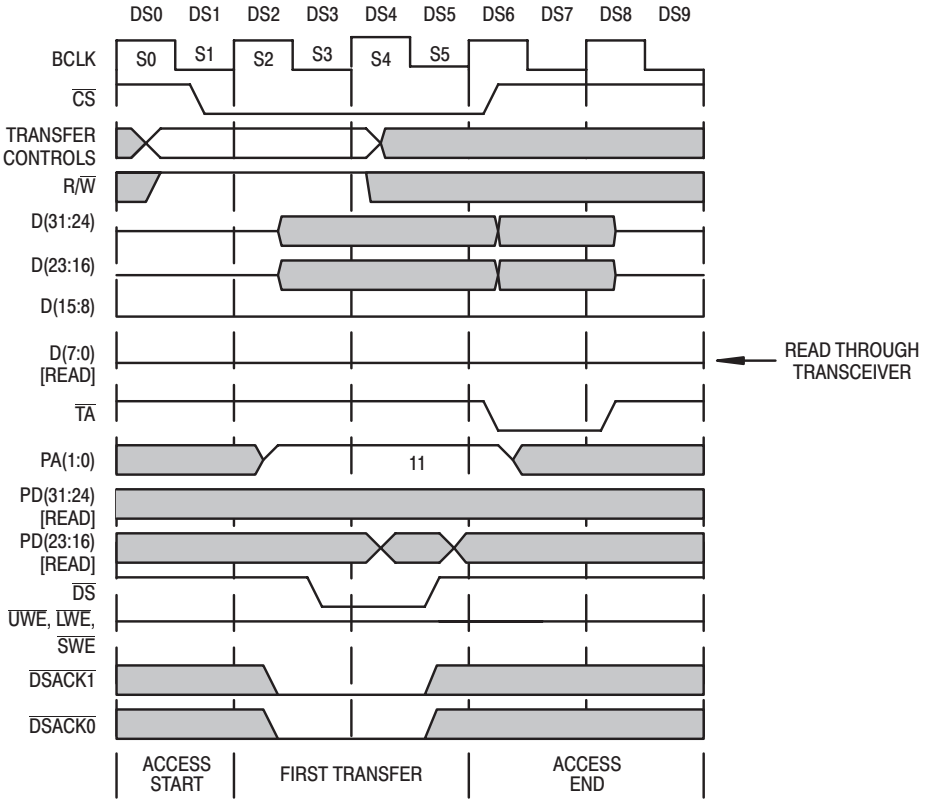


Figure 55. 8-Bit '040 READ From 32-Bit Peripheral Example (PA1,PA0=11)

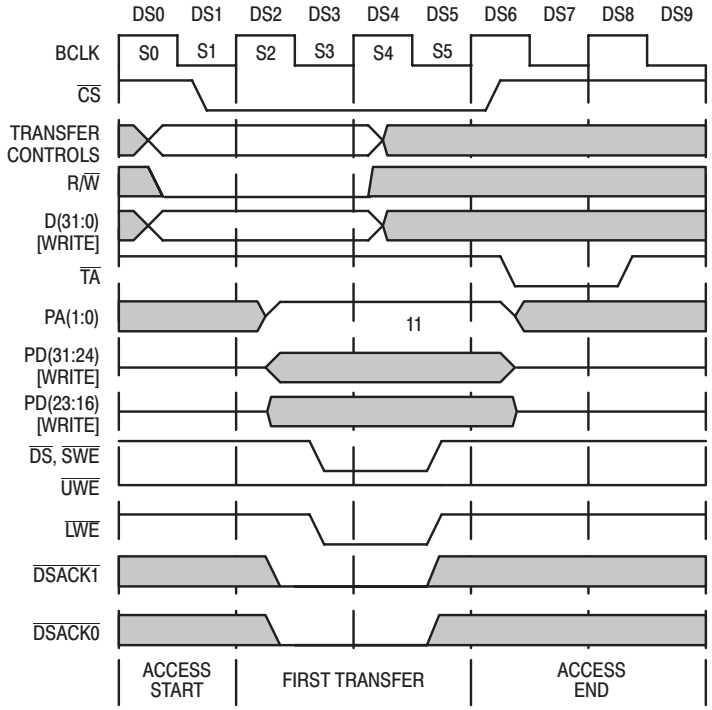
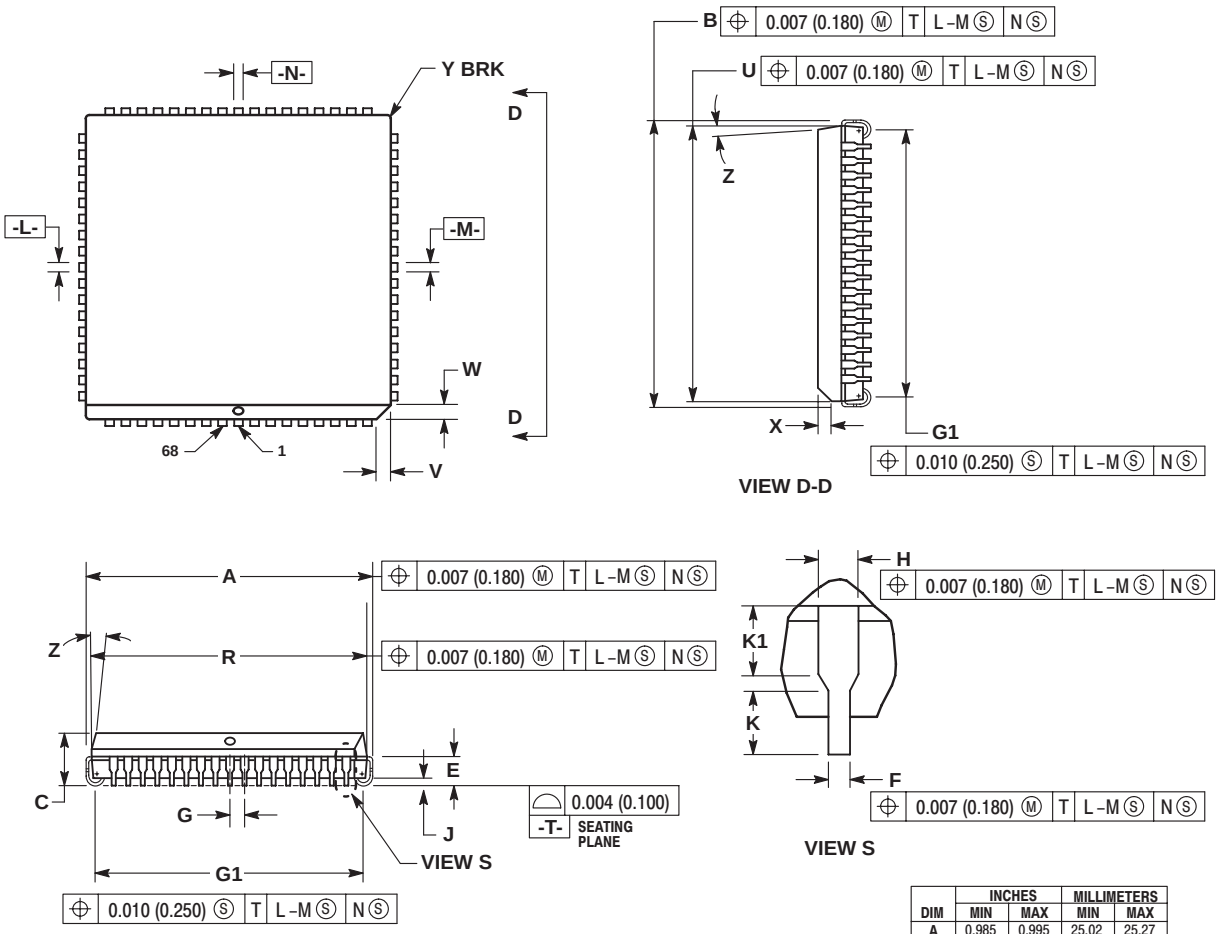


Figure 56. 8-Bit '040 WRITE to 32-Bit Peripheral Example (PA1,PA0=11)

OUTLINE DIMENSIONS

FN SUFFIX
PLASTIC PACKAGE
CASE 779-02
ISSUE C



DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.985	0.995	25.02	25.27
B	0.985	0.995	25.02	25.27
C	0.165	0.180	4.20	4.57
E	0.090	0.110	2.29	2.79
F	0.013	0.019	0.33	0.48
G	0.050 BSC		1.27 BSC	
H	0.026	0.032	0.66	0.81
J	0.020	—	0.51	—
K	0.025	—	0.64	—
R	0.950	0.956	24.13	24.28
U	0.950	0.956	24.13	24.28
V	0.042	0.048	1.07	1.21
W	0.042	0.048	1.07	1.21
X	0.042	0.056	1.07	1.42
Y	—	0.020	—	0.50
Z	2°	10°	2°	10°
G1	0.910	0.930	23.12	23.62
K1	0.040	—	1.02	—

NOTES:
1. DATUMS -L-, -M-, AND -N- DETERMINED WHERE TOP OF LEAD SHOULDER EXITS PLASTIC BODY AT MOLD PARTING LINE.
2. DIM G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
3. DIM R AND U DO NOT INCLUDE MOLD FLASH. ALLOWABLE MOLD FLASH IS 0.010 (0.250) PER SIDE.
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: INCH.
6. THE PACKAGE TOP MAY BE SMALLER THAN THE PACKAGE BOTTOM BY UP TO 0.012 (0.300). DIMENSIONS R AND U ARE DETERMINED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY EXCLUSIVE OF MOLD FLASH, TIE BAR BURRS, GATE BURRS AND INTERLEAD FLASH, BUT INCLUDING ANY MISMATCH BETWEEN THE TOP AND BOTTOM OF THE PLASTIC BODY.
7. DIMENSION H DOES NOT INCLUDE DAMBAR PROTRUSION OR INTRUSION. THE DAMBAR PROTRUSION(S) SHALL NOT CAUSE THE H DIMENSION TO BE GREATER THAN 0.037 (0.940). THE DAMBAR INTRUSION(S) SHALL NOT CAUSE THE H DIMENSION TO BE SMALLER THAN 0.025 (0.635).

Innovate with IDT and accelerate your future networks. Contact:

www.IDT.com

For Sales
800-345-7015
408-284-8200
Fax: 408-284-2775

For Tech Support
netcom@idt.com
480-763-2056

Corporate Headquarters
Integrated Device Technology, Inc.
6024 Silver Creek Valley Road
San Jose, CA 95138
United States
800 345 7015
+408 284 8200 (outside U.S.)

Asia Pacific and Japan
Integrated Device Technology
Singapore (1997) Pte. Ltd.
Reg. No. 199707558G
435 Orchard Road
#20-03 Wisma Atria
Singapore 238877
+65 6 887 5505

Europe
IDT Europe, Limited
Prime House
Barnett Wood Lane
Leatherhead, Surrey
United Kingdom KT22 7DE
+44 1372 363 339

www.IDT.com

XX-XXXX-XXXXX