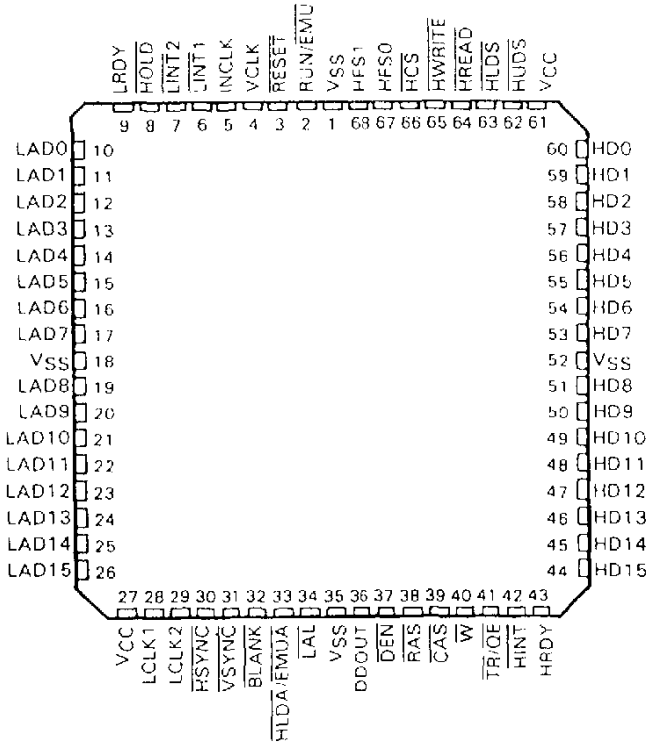
SPVS002C JUNE 1986 – REVISED JUNE 1991

- Instruction Cycle Time:
  - 132 ns . . . (TMS34010-60)
  - 160 ns . . . (TMS34010-50)
  - 200 ns . . . (TMS34010-40)
- Fully Programmable 32-Bit General-Purpose Processor with 128-Megabyte Address Range
- Pixel Processing, XY Addressing, and Window Checking Built into the Instruction Set
- Programmable 1, 2, 4, 8, or 16-Bit Pixel Size with 16 Boolean and 6 Arithmetic Pixel Processing Options (Raster-Ops)
- 30 General-Purpose 32-bit Registers and 32-bit Stack Pointer
- 256-Byte LRU On-Chip Instruction Cache
- Direct Interfacing to Both Conventional DRAM and Multiport Video RAM
- Dedicated 8/16-Bit Host Processor Interface and HOLD/HLDA Interface
- Programmable CRT Control ($\overline{HSYNC}$, $\overline{VSYNC}$, $\overline{BLANK}$)
- High-Level Language Support
- Full Line of Hardware and Software Development Tools Including a "C" Compiler
- 68-Leaded Packaging (PLCC)
- 5-Volt CMOS Technology

FN PACKAGE
(TOP VIEW)



description

The TMS34010 Graphics System Processor (GSP) is an advanced high-performance CMOS 32-bit microprocessor optimized for graphics display systems. With a built-in instruction cache, the ability to simultaneously access memory and registers, and an instruction set designed specifically for raster graphics operation, the TMS34010 provides user-programmable control of the CRT interface as well as the memory interface (both standard DRAM and multiport video RAM). The 1-gigabit address space is completely bit-addressable on bit boundaries using variable width data fields (1 to 32 bits). Additional graphics addressing modes support 1, 2, 4, 8, and 16-bit wide pixels. The TMS34010 is exceptionally well-supported by graphics software interface standards such as TIGA, MS-Windows, the X Window System, DGIS, and CGI, as well as a full line of hardware and software support tools. Current support is highlighted in the *TMS34010 Third Party Reference Guide* (literature number SPVB066C).

architecture

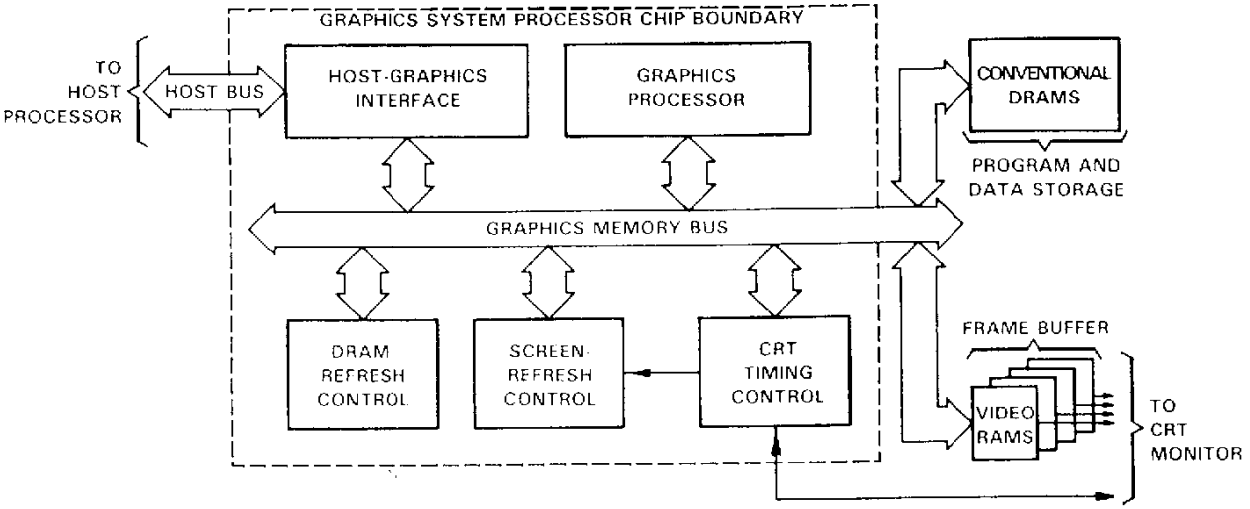
The TMS34010 is a CMOS 32-bit processor with hardware support for graphics operations such as PixBlts (raster ops) and curve-drawing algorithms. Also included is a complete set of general-purpose instructions with addressing tuned to support high-level languages. In addition to its ability to address a large external memory range, the TMS34010 contains 30 general-purpose 32-bit registers, a hardware stack pointer

and a 256-byte instruction cache. On-chip functions include 28 programmable I/O registers that contain CRT control, input/output control, and instruction parameters. The TMS34010 directly interfaces to dynamic RAMs and video RAMs and generates video monitor control signals. It also accommodates a conventional HOLD/HLDA shared access as well as a separate, generalized interface for communicating with any standard host processor.

pin descriptions

PIN NAME	PIN NUMBER	I/O	DESCRIPTION
			Host Interface Bus Pins
$\overline{HCS}$	66	I	Host chip select
HD0-HD15	44-51, 53-60	I/O	Host bidirectional data bus
HFS0, HFS1	67, 68	I	Host function select
$\overline{HINT}$	42	O	Host interrupt request
$\overline{HLDS}$	63	I	Host lower data select
$\overline{HUDS}$	62	I	Host upper data select
HRDY	43	O	Host ready
$\overline{HREAD}$	64	I	Host read strobe
$\overline{HWRITE}$	65	I	Host write strobe
			Local Bus Interface Pins
$\overline{RAS}$	38	O	Local row-address strobe
$\overline{CAS}$	39	O	Local column-address strobe
DDOUT	36	O	Local data direction out
$\overline{DEN}$	37	O	Local data enable
LAD0-LAD15	10-17, 19-26	I/O	Local address/data bus
$\overline{LAI}$	34	O	Local address latched
LCLK1, LCLK2	28, 29	O	Local output clocks
$\overline{LINT1}$, $\overline{LINT2}$	6, 7	I	Local interrupt request pins
LRDY	9	I	Local ready
$\overline{TR/OE}$	41	O	Local shift register transfer or output enable
$\overline{W}$	40	O	Local write strobe
NCLK	5	I	Input clock
			Hold and Emulation
$\overline{HOLD}$	8	I	Hold request
RUN/$\overline{EMU}$	2	I	Run/Not emulate
$\overline{HLDA/EMUA}$	33	O	Hold acknowledge or emulate acknowledge
			Video Timing Signals
$\overline{BLANK}$	32	O	Blanking
$\overline{HSYNC}$	30	I/O	Horizontal sync
VCLK	4	I	Video clock
$\overline{VSYNC}$	31	I/O	Vertical sync
			Miscellaneous
$\overline{RESET}$	3	I	Reset
V_{CC}	27, 61	I	Nominal 5-volt power supply
V_{SS}	1, 18, 35, 52	I	Ground

system block diagram



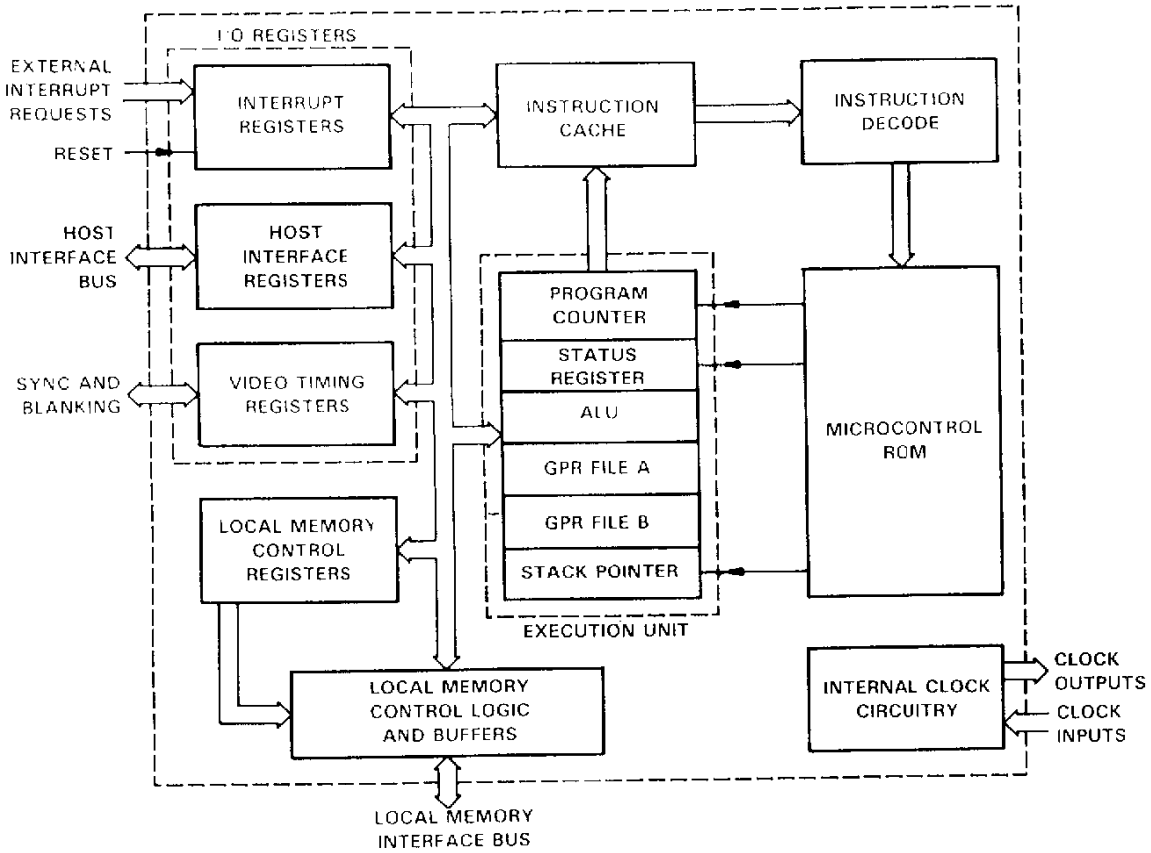


FIGURE 1. TMS34010 INTERNAL ARCHITECTURE

The TMS34010 provides single-cycle execution of most common integer arithmetic and Boolean operations from its instruction cache. Additionally, the TMS34010 incorporates a hardware barrel shifter that provides a single state bidirectional shift and rotate function for 1 to 32 bits.

A microcoded local memory controller supports pipelined memory write operations of variable-size fields that can be performed in parallel with subsequent instruction execution.

TMS34010 graphics processing hardware supports pixel and pixel-array processing capabilities for both monochrome and color systems that have a variety of pixel sizes. The hardware incorporates two-operand raster operations with Boolean and arithmetic operations, XY addressing, window clipping, window checking operations, 1 to n bits per pixel transforms, transparency, and plane masking. The architecture further supports operations on single pixels (PIXT instructions) or on two-dimensional pixel arrays of arbitrary size (PixBlts).

The TMS34010's flexible graphics processing capabilities allow software-based graphics algorithms without sacrificing performance. These algorithms include: arbitrary window size, custom incremental curve drawing, and two-operand raster operations.

register files

Boolean, arithmetic, byte, and field move instructions operate on data within the TMS34010's general-purpose register files. The TMS34010 contains thirty-one 32-bit registers, including a system stack pointer (SP). The SP is accessible to both Register File A and B as the sixteenth register. Transfers between registers and memory are facilitated via a complete set of field MOVE instructions with selectable field sizes. Transfers between registers are facilitated via the MOVE instruction.

The fifteen general-purpose registers in Register File A are used for high-level language support and assembly language programming. The fifteen registers in Register File B are dedicated to special functions during PixBlts and other pixel operations, but can be used as general-purpose registers at other times.

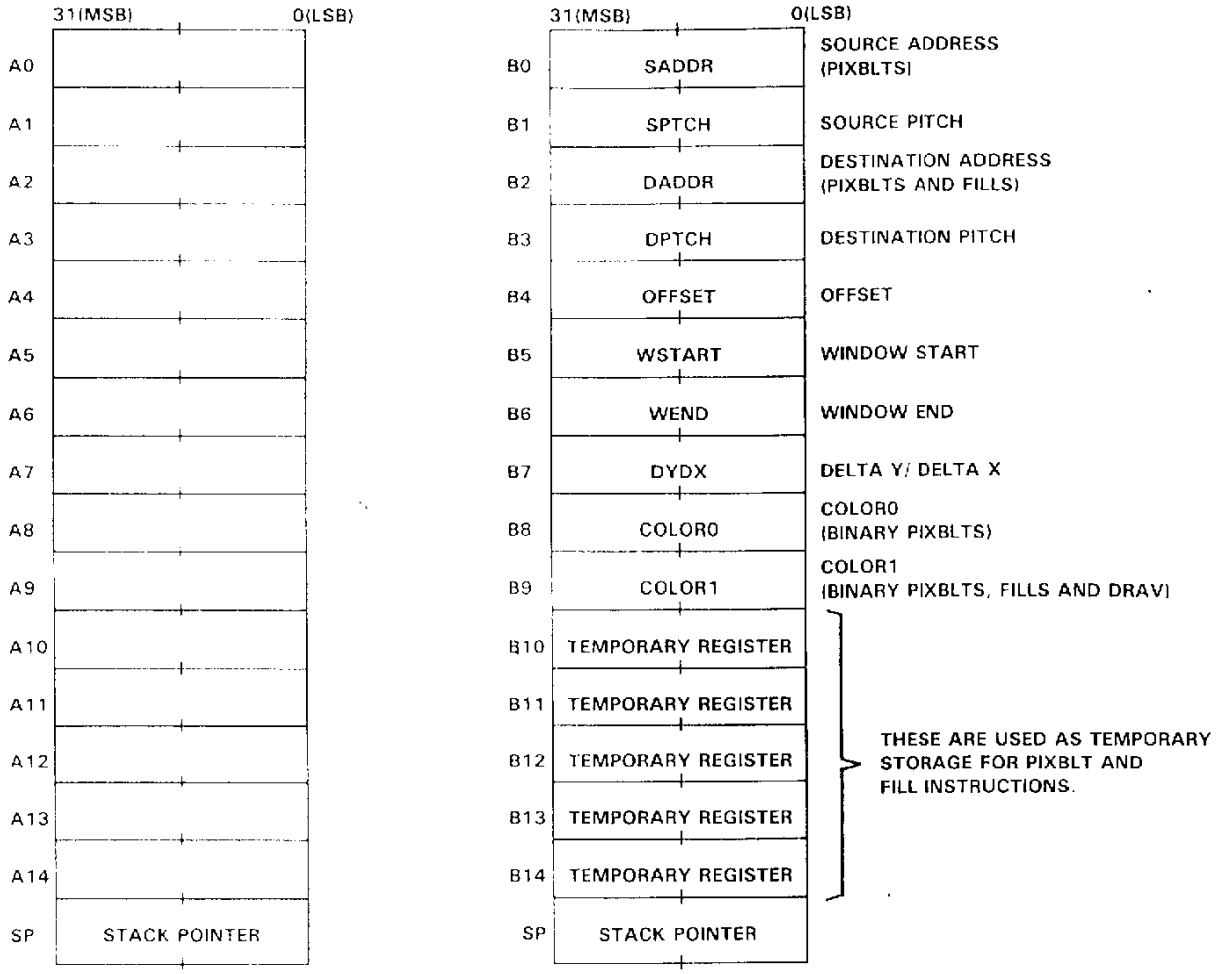


FIGURE 2. REGISTER FILES A AND B

program counter (PC)

The TMS34010's 32-bit program counter register points to the next instruction-stream word to be fetched. Since instruction words are aligned to 16-bit boundaries, the four LSBs of the PC are always zero.

instruction cache

An on-chip instruction cache contains 256 bytes of RAM and provides fast access to instructions. It operates automatically and is transparent to software. The cache is divided into four 64-byte segments. Associated with each segment is a 23-bit segment address register to identify the addresses in memory corresponding to the current contents of the cache segment. Each cache segment is further partitioned into eight subsegments of four words each. Each subsegment has associated with it a present (P) flag to indicate whether the subsegment contains valid data.

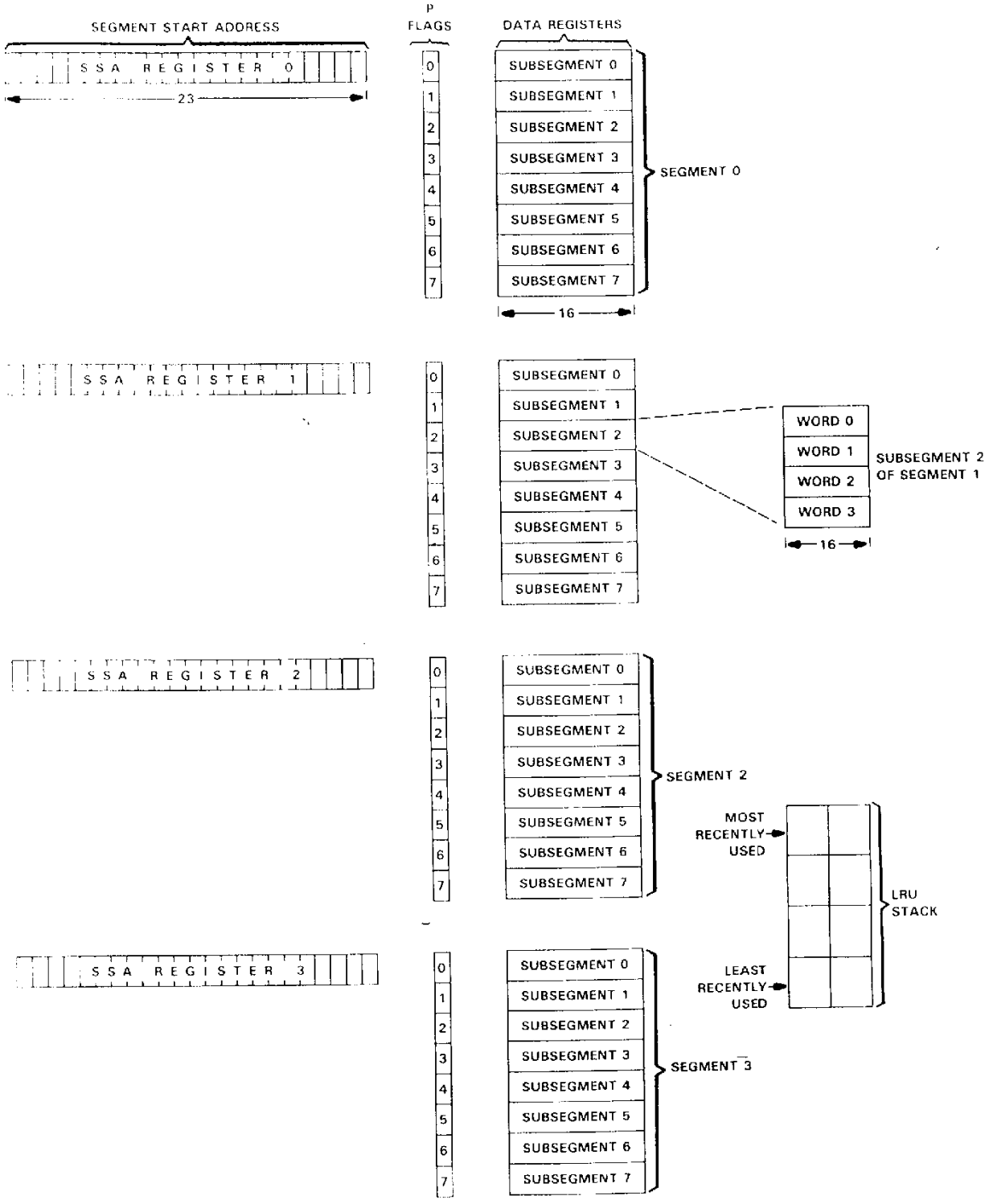
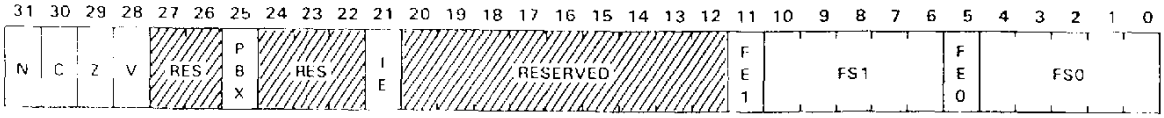


FIGURE 3. INSTRUCTION CACHE

The cache is loaded only when an instruction requested by the TMS34010 is not already contained within the cache. A least-recently-used (LRU) algorithm is used to determine which of the four segments of the cache is overwritten with the new data. For this purpose, an internal four-by-two LRU stack is used to keep track of cache usage.

status register

The status register (ST) is a special-purpose 32-bit register dedicated to status codes set by the results of implicit and explicit compare operations and parameters used to specify the length and behavior of fields 0 and 1.



N	Sign bit	IE	Interrupt enable bit
C	Carry bit	FE1	Field extension bit 1
Z	Zero bit	FS1	Field size bit 1
V	Overflow bit	FE0	Field extension bit 0
PBX	PixBlt executing	FS0	Field size bit 0

FIGURE 4. STATUS REGISTER

fields, bytes, pixels, and pixel arrays

A 26-bit address output by the TMS34010 selects a 16-bit word of physical memory; logically, however, the TMS34010 views memory data as fields addressable at the bit level. Primitive data types supported by the TMS34010 include: bytes, pixels, two 1- to 32-bit fields, and user-defined pixel arrays.

Fields 0 and 1 are specified independently to be from 1 to 32 bits in length. Bytes are special 8-bit cases of the field data type, while pixels are 1, 2, 4, 8 or 16 bits in length. In general, fields (including bytes) may start and terminate on arbitrary bit boundaries; pixels must pack evenly into 16-bit words.

pixel operations

Pixel arrays are two-dimensional data types of user-defined width, height, pixel depth (number of bits per pixel), and pitch (distance between rows). A pixel or pixel array may be accessed by means of either its memory address or its XY coordinates. Transfers of individual pixels or pixel blocks are influenced by the pixel processing, transparency, window checking, plane masking, or corner adjust operations selected.

I/O registers

The GSP contains an on-chip block of twenty-eight 16-bit I/O registers mapped into the TMS34010's memory address space. They can be accessed either by the TMS34010's CPU or by the host processor via the host interface. The I/O registers contain control parameters necessary to configure the operation of the following interfaces: interface to host processor (5 I/O registers), interface to local memory (6 registers), video timing and screen refresh functions (15 registers), and externally and internally generated interrupts (2 registers). The I/O registers also furnish status information on these interfaces.

ADDRESS	REGISTER	
0C00001F0h	REFCNT	DRAM REFRESH COUNT
0C00001E0h	DPYADR	DISPLAY ADDRESS
0C00001D0h	VCOUNT	VERTICAL COUNT
0C00001C0h	HCOUNT	HORIZONTAL COUNT
0C00001B0h	DPYTAP	DISPLAY TAP POINT
0C00001A0h		
0C0000190h	RESERVED	
0C0000180h		
0C0000170h		
0C0000160h	PMASK	PLANE MASK
0C0000150h	PSIZE	PIXEL SIZE
0C0000140h	CONVDP	CONVERSION (DESTINATION PITCH)
0C0000130h	CONVSP	CONVERSION (SOURCE PITCH)
0C0000120h	INTPEND	INTERRUPT PENDING
0C0000110h	INTENB	INTERRUPT ENABLE
0C0000100h	HSTCTLH	HOST CONTROL (8 MSB'S)
0C00000F0h	HSTCTLL	HOST CONTROL (8 LSB'S)
0C00000E0h	HSTADRH	HOST ADDRESS (16 MSB'S)
0C00000D0h	HSTADRL	HOST ADDRESS (16 LSB'S)
0C00000C0h	HSTDATA	HOST DATA
0C00000B0h	CONTROL	CONTROL
0C00000A0h	DPYINT	DISPLAY INTERRUPT
0C0000090h	DPYSTRT	DISPLAY START
0C0000080h	DPYCTL	DISPLAY CONTROL
0C0000070h	VTOTAL	VIDEO TOTAL
0C0000060h	VSBLNK	VERTICAL START BLANK
0C0000050h	VEBLNK	VERTICAL END BLANK
0C0000040h	VESYNC	VERTICAL END SYNC
0C0000030h	HTOTAL	HORIZONTAL TOTAL
0C0000020h	HSBLNK	HORIZONTAL START BLANK
0C0000010h	HEBLNK	HORIZONTAL END BLANK
0C0000000h	HESYNC	HORIZONTAL END SYNC

FIGURE 5. I/O REGISTERS

host interface registers

The host interface registers are provided for communications between the TMS34010 and the host processor. The registers are mapped into five of the I/O register locations accessible to the TMS34010. These same registers are mapped into four locations in the GSP interface to the host.

One of the registers is devoted to host interface control functions such as the passing of interrupt requests and 3-bit status codes from host to TMS34010 and from TMS34010 to host. Other control functions available to the host processor include flushing the instruction cache, halting the TMS34010, and transmitting a non maskable interrupt request to the TMS34010.

The remaining host registers are used for block transfers between the TMS34010 and host processor. The host uses these registers to indirectly access blocks within the TMS34010's local memory. Two of the 16-bit registers contain the 32-bit address of the current word location in memory. Another 16-bit register buffers data transferred to and from the memory by the host processor. The host interface can be programmed to automatically increment the pointer address following each transfer to provide the host with rapid access to a block of sequential addresses.

memory interface control registers

Six of the I/O registers are dedicated to various local memory interface functions including:

- Frequency and type of DRAM refresh cycles
- Pixel size
- Masking (write protection) of individual color planes
- Various pixel access control parameters
  - Window checking mode
    Boolean or arithmetic pixel processing operation
  - Transparency
  - PixBlt direction control

video timing and screen refresh

Fourteen I/O registers are dedicated to video timing and screen refresh functions. The TMS34010 generates the horizontal sync ($\overline{\text{HSYNC}}$), vertical sync ($\overline{\text{VSYNC}}$), and blanking ($\overline{\text{BLANK}}$) signals used to drive a video monitor in a graphics system. These signals are controlled by means of a set of programmable video timing I/O registers and are based on the input video clock, VCLK. VCLK does not have to be synchronous with respect to INCLK, the TMS34010's CPU input clock.

The TMS34010 directly supports multiport video RAMs (VRAMs) by generating the memory-to-register load cycles necessary to refresh the display being shown on the video monitor. The memory locations from which display information is taken, as well as the number of horizontal scan lines displayed between memory-to-register load cycles, are programmable. VRAM tap point addresses are also fully programmable to support horizontal panning.

The TMS34010 supports various screen resolutions and either interlaced or noninterlaced video. The TMS34010 can optionally be programmed to synchronize to externally generated sync signals so that graphics images created by the TMS34010 can be superimposed upon images created externally. The external sync mode can also be used to synchronize the video signals generated by two or more TMS34010 chips in a multiple-TMS34010 graphics system.

interrupt interface registers

Two dedicated I/O registers monitor and mask interrupt requests to the TMS34010, including two externally generated interrupts and three internally generated interrupts. An internal interrupt request can be generated on one of the following conditions:

- Window violation: an attempt has been made to write a pixel to a location inside or outside a specified window boundary.
- Host interrupt: the host processor has set the interrupt request bit in the host control register.
- Display interrupt: a specified line number in the frame has been displayed on the screen.

A nonmaskable interrupt occurs when the host processor sets a particular control bit in the host interface registers. The TMS34010 reset function is controlled by a dedicated pin.

memory controller/local memory interface

The memory controller manages the TMS34010's interface to the local memory and automatically performs the bit alignment and masking necessary to access data located at arbitrary bit boundaries within memory. The memory controller operates autonomously with respect to the CPU. It has a "write queue" one field (1 to 32 bits) deep that permits it to complete the memory cycles necessary to insert the field into memory without delaying the execution of subsequent instructions. Only when a second memory operation is required before the memory controller has completed the first operation is the TMS34010 forced to defer instruction execution.

The TMS34010 directly interfaces to all standard dynamic RAMs and, in particular to JEDEC standard 64K and 256K video RAMs such as the TMS4161 and TMS4461 Multiport VRAMs. The TMS34010 memory interface consists of a triple-multiplexed address/data bus plus the associated control signals. Row address, column address, and data are multiplexed over the same address/data lines. DRAM refresh is supported with a variety of modes including $\overline{CAS}$ before $\overline{RAS}$ refresh.

TMS34010 memory map

From the programmer's point of view, the TMS34010 treats data and instructions as residing in the same memory space.

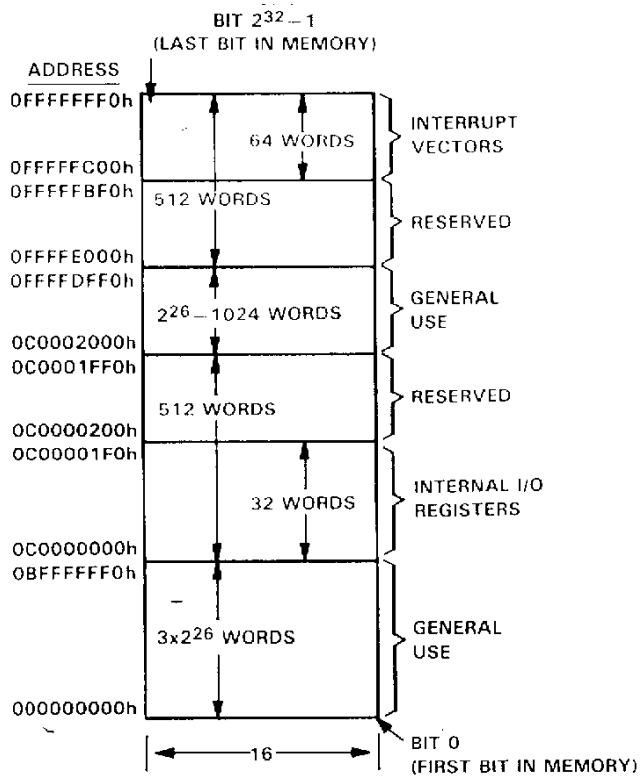


FIGURE 6. MEMORY MAP

instruction set

The TMS34010 instructions fall into three categories. The *graphics instructions* manipulate pixel data, accessed via memory addresses or XY coordinates. They provide support for graphics operations such as array and raster ops, pixel processing, windowing, plane masking, and transparency. The *move instructions* comprehend bit addressing and field operations; they manipulate fields of data using linear addressing for transfer to and from memory and the register file. The TMS34010 *general-purpose instructions* provide a complete set of arithmetic and Boolean operations on the register file as well as general program control and data processing. Partial timing information is provided in the table below. The two values given for jump instructions in the Minimum Cycles column indicate the jump and no-jump conditions, respectively. Full timing information can be obtained in the *TMS34010 User's Guide* (number SPVU001B).

The following abbreviations are used below in the opcodes: S (source register), D (destination register), R (register file select), F (field select), K (constant), M (cross A/B file boundary), Z (draw option), code (jump select code), X (don't care), N (trap select and stack adjust), RS (source register), RD (destination register), xxxx (address displacement), IL (32-bit immediate operand), and IW (16-bit immediate operand).

GRAPHICS INSTRUCTIONS

SYNTAX	DESCRIPTION	NO. WORDS	MINIMUM CYCLES	16-BIT OPCODE MSB			LSB	STATUS BITS			
ADDXY Rs, Rd	Add Registers in XY Mode	1	1	1110	000S	SSSR	DDDD	N	C	Z	V
CMPXY Rs, Rd	Compare X and Y Halves of Registers	1	3	1110	010S	SSSR	DDDD	N	C	Z	V
CPW Rs, Rd	Compare Point to Window	1	1	1110	011S	SSSR	DDDD	–	–	–	V
CVXYL Rs, Rd	Convert XY Address to Linear Address	1	3	1110	100S	SSSR	DDDD	–	–	–	–
DRAV Rs, Rd	Draw and Advance	1	†	1111	011S	SSSR	DDDD	–	–	–	V
FILL L	Fill Array with Processed Pixels: Linear	1	†	0000	1111	1100	0000	–	–	–	–
FILL XY	Fill Array with Processed Pixels: XY	1	†	0000	1111	1110	0000	–	–	–	V
LINE Z	Line Draw	1	†	1101	1111	Z001	1010	–	–	–	V
MOVX Rs, Rd	Move X Half of Register	1	1	1110	110S	SSSR	DDDD	–	–	–	–
MOVY Rs, Rd	Move Y Half of Register	1	1	1110	111S	SSSR	DDDD	–	–	–	–
PIXBLT B,L	Pixel Block Transfer: Binary to Linear	1	†	0000	1111	1000	0000	–	–	–	–
PIXBLT B,XY	Pixel Block Transfer and Expand: Binary to XY	1	†	0000	1111	1010	0000	–	–	–	V
PIXBLT L,L	Pixel Block Transfer: Linear to Linear	1	†	0000	1111	0000	0000	–	–	–	–
PIXBLT L,XY	Pixel Block Transfer: Linear to XY	1	†	0000	1111	0010	0000	–	–	–	V
PIXBLT XY, L	Pixel Block Transfer: XY to Linear	1	†	0000	1111	0100	0000	–	–	–	–
PIXBLT XY,XY	Pixel Block Transfer: XY to XY	1	†	0000	1111	0110	0000	–	–	–	V
PIXT Rs,*Rd	Pixel Transfer: Register to Indirect	1	†	1111	100S	SSSR	DDDD	–	–	–	–
PIXT Rs,*Rd.XY	Pixel Transfer: Register to Indirect XY	1	†	1111	000S	SSSR	DDDD	–	–	–	V
PIXT *Rs, Rd	Pixel Transfer: Indirect to Register	1	4	1111	101S	SSSR	DDDD	–	–	–	–
PIXT *Rs,*Rd	Pixel Transfer: Indirect to Indirect	1	†	1111	110S	SSSR	DDDD	–	–	–	–
PIXT *Rs.XY, Rd	Pixel Transfer: Indirect XY to Register	1	6	1111	001S	SSSR	DDDD	–	–	–	–
PIXT *Rs.XY,*Rd.XY	Pixel Transfer: Indirect XY to Indirect XY	1	†	1111	010S	SSSR	DDDD	–	–	–	V
SUBXY Rs,Rd	Subtract Registers in XY Mode	1	1	1110	001S	SSSR	DDDD	N	C	Z	V

†Number of cycles depends on pixel size and/or pixel array size and graphics option selected. See *TMS34010 User's Guide* (SPVU001B)

MOVE INSTRUCTIONS

SYNTAX	DESCRIPTION	NO. WORDS	MINIMUM CYCLES	16-BIT OPCODE MSB LSB	STATUS BITS
MOVB Rs,*Rd	Move Byte: Register to Indirect	1	†	1000 110S SSSR DDDD	– – – –
MOVB *Rs,Rd	Move Byte: Indirect to Register	1	†	1000 111S SSSR DDDD	N – Z 0
MOVB *Rs,*Rd	Move Byte: Indirect to Indirect	1	†	1001 110S SSSR DDDD	– – – –
MOVB Rs,*Rd(offset)	Move Byte: Register to Indirect with offset.	2	†	1010 110S SSSR DDDD	– – –
MOVB *Rs(offset),Rd	Move Byte: Indirect with offset. to Register	2	†	1010 111S SSSR DDDD	N – Z 0
MOVB *Rs(offset),*Rd(offset)	Move Byte: Ind. with offset. to Ind. with offset	3	†	1011 110S SSSR DDDD	– – – –
MOVB Rs,@Daddress	Move Byte: Register to Absolute	3	†	0000 0101 111R SSSS	– – – –
MOVB @Saddress,Rd	Move Byte: Absolute to Register	3	†	0000 0111 111R DDDD	N – Z 0
MOVB @Saddress,@Daddress	Move Byte: Absolute to Absolute	5	†	0000 0011 0100 0000	– – –
MOVE Rs,Rd	Move Register to Register	1	1	0100 11MS SSSR DDDD	N – Z 0
MOVE Rs,*Rd,F	Move Field: Register to Indirect	1	†	1000 00FS SSSR DDDD	– – –
MOVE Rs,*Rd,F	Move Field: Register to Indirect (pre-dec)	1	†	1010 00FS SSSR DDDD	– – –
MOVE Rs,*Rd+,F	Move Field: Register to Indirect (post-inc)	1	†	1001 00FS SSSR DDDD	– – – –
MOVE *Rs,Rd,F	Move Field: Indirect to Register	1	†	1000 01FS SSSR DDDD	N – Z 0
MOVE *Rs,Rd,F	Move Field: Indirect (pre-dec) to Register	1	†	1010 01FS SSSR DDDD	N – Z 0
MOVE *Rs+,Rd,F	Move Field: Indirect (post-inc) to Register	1	†	1001 01FS SSSR DDDD	N – Z 0
MOVE *Rs,*Rd,F	Move Field: Indirect to Indirect	1	†	1000 10FS SSSR DDDD	– – – –
MOVE *Rs,*Rd,F	Move Field: Ind. (pre-dec) to Ind. (pre-dec)	1	†	1010 10FS SSSR DDDD	– – – –
MOVE *Rs+,*Rd+,F	Move Field: Ind. (post-inc) to Ind. (post-inc)	1	†	1001 10FS SSSR DDDD	– –
MOVE Rs,*Rd(offset),F	Move Field: Register to Indirect with offset.	2	†	1011 00FS SSSR DDDD	– –
MOVE *Rs(offset),Rd,F	Move Field: Indirect with offset. to Register	2	†	1011 01FS SSSR DDDD	N – Z 0
MOVE *Rs(offset),*Rd+,F	Move Field: Ind. with offset. to Ind. (post-inc)	2	†	1101 00FS SSSR DDDD	– – – –
MOVE *Rs(offset),*Rd(offset),F	Move Field: Ind. with offset. to Ind. with offset.	3	†	1011 10FS SSSR DDDD	– – – –
MOVE Rs,@Daddress,F	Move Field: Register to Absolute	3	†	0000 01F1 100R SSSS	– – – –
MOVE @Saddress,Rd,F	Move Field: Absolute to Register	3	†	0000 01F1 101R DDDD	N – Z 0
MOVE @Saddress,*Rd+,F	Move Field: Absolute to Indirect (post-inc)	3	†	1101 01F0 000R DDDD	– – –
MOVE @Saddress,@Daddress,F	Move Field: Absolute to Absolute	5	†	0000 01F1 1100 0000	– – –

†Number of cycles depends on field size and alignment. See *TMS34010 User's Guide* (SPVU001B).

GENERAL INSTRUCTIONS

SYNTAX	DESCRIPTION	NO. WORDS	MINIMUM CYCLES	16-BIT OPCODE MSB LSB	STATUS BITS
ABS Rd	Store Absolute Value	1	1	0000 0011 100R DDDD	N – Z 0
ADD Rs,Rd	Add Registers	1	1	0100 000S SSSR DDDD	N C Z V
ADDC Rs,Rd	Add Register with Carry	1	1	0100 001S SSSR DDDD	N C Z V
ADDI IW,Rd	Add Immediate (16 Bits)	2	2	0000 1011 000R DDDD	N C Z V
ADDI IL,Rd	Add Immediate (32 Bits)	3	3	0000 1011 001R DDDD	N C Z V
ADDK K,Rd	Add Constant (5 Bits)	1	1	0001 00KK KKKR DDDD	N C Z V
AND Rs,Rd	AND Registers	1	1	0101 000S SSSR DDDD	– – Z –
ANDI IL,Rd	AND Immediate (32 Bits)	3	3	0000 1011 100R DDDD	– – Z –
ANDN Rs,Rd	AND Register with Complement	1	1	0101 001S SSSR DDDD	– – Z –
ANDNI IL,Rd	AND Not Immediate (32 Bits)	3	3	0000 1011 100R DDDD	– – Z –
BTST K,Rd	Test Register Bit – Constant	1	1	0001 11KK KKKR DDDD	– – Z –
BTST Rs,Rd	Test Register Bit – Register	1	2	0100 101S SSSR DDDD	– – Z –
CLR Rd	Clear Register	1	1	0101 011D DDDR DDDD	– –
CLRC	Clear Carry	1	1	0000 0011 0010 0000	– 0 –
CMP Rs,Rd	Compare Registers	1	1	0100 100S SSSR DDDD	N C Z V
CMPI IW,Rd	Compare Immediate (16 Bits)	2	2	0000 1011 010R DDDD	N C Z V
CMPI IL,Rd	Compare Immediate (32 Bits)	3	3	0000 1011 011R DDDD	N C Z V
DEC Rd	Decrement Register	1	1	0001 0100 001R DDDD	– – – –
DINT	Disable Interrupts	1	3	0000 0011 0110 0000	– – – –
DIVS Rs,Rd	Divide Registers Signed	1	40	0101 100S SSSR DDDD	N – Z V
DIVU Rs,Rd	Divide Registers Unsigned	1	37	0101 101S SSSR DDDD	– – Z V
EINT	Enable Interrupts	1	3	0000 1101 0110 0000	– – – –
EXGF Rd,F	Exchange Field Size	1	1	1101 01F1 000R DDDD	– – – –
INC Rd	Increment Register	1	1	0001 0000 001R DDDD	– – – –
LMO Rs,Rd	Leftmost One	1	1	0110 101S SSSR DDDD	– – Z –
MMFM Rs,Register List	Move Multiple Registers from Memory	2	†	0000 1001 101R DDDD	– – – –
MMTM Rd,Register List	Move Multiple Registers to Memory	2	†	0000 1001 100R DDDD	– – – –
MODS Rs,Rd	Modulus Signed	1	40	0110 110S SSSR DDDD	N – Z V
MODU Rs,Rd	Modulus Unsigned	1	35	0110 111S SSSR DDDD	– – Z V
MOVI IW,Rd	Move Immediate (16 Bits)	2	2	0000 1001 110R DDDD	N – Z 0
MOVI IL,Rd	Move Immediate (32 Bits)	3	3	0000 1001 111R DDDD	N – Z 0
MOVK K,Rd	Move Constant (5 Bits)	1	1	0001 10KK KKKR DDDD	– – – –
MPYS Rs,Rd	Multiply Registers (Signed)	1	$5+\frac{FS1}{2}$	0101 110S SSSR DDDD	N – Z –
MPYU Rs,Rd	Multiply Registers (Unsigned)	1	$5+\frac{FS1}{2}$	0101 111S SSSR DDDD	– – Z –
NEG Rd	Negate Register	1	1	0000 0011 101R DDDD	N C Z V
NEGB Rd	Negate Register with Borrow	1	1	0000 0011 110R DDDD	N C Z V
NOP	No operation	1	1	0000 0011 0000 0000	– – – –
NOT Rd	Complement Register	1	1	0000 0011 111R DDDD	– – Z –
OR Rs,Rd	OR Registers	1	1	0101 010S SSSR DDDD	– – Z –
ORI IL,Rd	OR Immediate (32 bits)	3	3	0000 1011 101R DDDD	– – Z –
RL K,Rd	Rotate Left – Constant	1	1	0011 00KK KKKR DDDD	– C Z –
RL Rs,Rd	Rotate Left – Register	1	1	0110 100S SSSR DDDD	– C Z –
SETC	Set Carry	1	1	0000 1101 1110 0000	– 1 – –
SETF FS,FE,F	Set Field Parameters	1	1,2	0000 01F1 01FS SSSS	– – – –
SEXT Rd,F	Sign Extend to Long	1	3	0000 01F1 000R DDDD	N – Z –

†Number of cycles depends on number of registers in list and stack alignment. See TMS34010 User's Guide (SPVU001B).

SYNTAX	DESCRIPTION	NO. WORDS	MINIMUM CYCLES	16-BIT OPCODE MSB			LSB	STATUS BITS			
SLA K,Rd	Shift Left Arithmetic - Constant	1	3	0010	00KK	KKKR	DDDD	N	C	Z	V
SLA Rs,Rd	Shift Left Arithmetic - Register	1	3	0110	000S	SSSR	DDDD	N	C	Z	V
SLL K,Rd	Shift Left Logical - Constant	1	1	0010	01KK	KKKR	DDDD	–	C	Z	–
SLL Rs,Rd	Shift Left Logical - Register	1	1	0110	001S	SSSR	DDDD	–	C	Z	–
SRA K,Rd	Shift Right Arithmetic - Constant	1	1	0010	10KK	KKKR	DDDD	N	C	Z	–
SRA Rs,Rd	Shift Right Arithmetic - Register	1	1	0110	010S	SSSR	DDDD	N	C	Z	–
SRL K,Rd	Shift Right Logical - Constant	1	1	0010	11KK	KKKR	DDDD		C	Z	–
SRL Rs,Rd	Shift Right Logical - Register	1	1	0110	011S	SSSR	DDDD	–	C	Z	–
SUB Rs,Rd	Subtract Registers	1	1	0100	010S	SSSR	DDDD	N	C	Z	V
SUBB Rs,Rd	Subtract Registers with Borrow	1	1	0100	011S	SSSR	DDDD	N	C	Z	V
SUBI IW,Rd	Subtract Immediate (16 Bits)	2	2	0000	1011	111R	DDDD	N	C	Z	V
SUBI IL,Rd	Subtract Immediate (32 Bits)	3	3	0000	1101	000R	DDDD	N	C	Z	V
SUBK K,Rd	Subtract Immediate (5 Bits)	1	1	0001	01KK	KKKR	DDDD	N	C	Z	V
XOR Rs,Rd	Exclusively OR Registers	1	1	0101	011S	SSSR	DDDD	–	–	Z	–
XORI IL,Rd	Exclusively OR Immediate Value (32 Bits)	3	3	0000	1011	110D	DDDD	–	–	Z	–
ZEXT Rd,F	Zero Extend to Long	1	1	0000	01F1	001R	DDDD	–	–	Z	–

PROGRAM CONTROL AND CONTEXT SWITCHING

SYNTAX	DESCRIPTION	NO. WORDS	MINIMUM CYCLES†	16-BIT OPCODE MSB			LSB	STATUS BITS			
CALL Rs	Call Subroutine Indirect	1	6	0000	1001	001R	DDDD	–	–	–	–
CALLA Address	Call Subroutine Absolute	3	6	0000	1101	0101	1111	–	–	–	–
CALLR Address	Call Subroutine Relative	2	5	0000	1101	0011	1111	–	–	–	–
DSJ Rd,Address	Decrement Register and Skip Jump	2	3,2	0000	1101	100R	DDDD	–	–	–	–
DSJEQ Rd,Address	Conditionally Decrement Register and Skip Jump	2	3,2	0000	1101	101R	DDDD	–	–	–	–
DSJNE Rd,Address	Conditionally Decrement Register and Skip Jump	2	3,2	0000	1101	110R	DDDD	–	–	–	–
DSJS Rd,Address	Decrement Register and Skip Jump - Short	1	2,3	0011	1Dxx	xxxR	DDDD	–	–	–	–
EMU	Initiate Emulation	1	6	0000	0001	0000	0000	–	–	–	–
EXGPC Rd	Exchange Program Counter with Register	1	2	0000	0001	001R	DDDD		–	–	–
GETPC Rd	Get Program Counter into Register	1	1	0000	0001	010R	DDDD		–	–	–
GETST Rd	Get Status Register into Register	1	1	0000	0001	100R	DDDD	–	–	–	–
JAcc Address	Jump Absolute Conditional	3	3,4	1100	code	1000	0000	–			
JRcc Address	Jump Relative Conditional	2	3,2	1100	code	0000	0000	–	–	–	–
JRcc Address	Jump Relative Conditional - Short	1	2,1	1100	code	xxxx	xxxx	–	–	–	
JUMP Rs	Jump Indirect	1	2	0000	0001	011R	DDDD	–	–	–	–
POPST	Pop Status Register from Stack	1	8	0000	0001	1100	0000	–	–	–	–
PUSHST	Push Status Register onto Stack	1	2	0000	0001	1110	0000	–	–	–	–
PUTST Rs	Copy Register into Status	1	3	0000	0001	101R	DDDD	N	C	Z	V
RETI	Return from Interrupt	1	11	0000	1001	0100	0000	N	C	Z	V
RETS (N)	Return from Subroutine	1	7	0000	1001	011N	NNNN	–	–	–	–
REV Rd	Get Revision Number	1	1	0000	0000	001R	DDDD	–	–	–	–
TRAP N	Software Interrupt	1	16	0000	1001	000N	NNNN	0	0	0	0

† Where two numbers appear, the first number assumes that the jump is taken, and the second assumes that the jump is not taken.

hardware and software support for the TMS34010

PART NUMBER	DESCRIPTION
TMS340SDK-PC	**TMS340 Software Developer's Kit** The SDK meets the needs of those developers of TIGA-compatible software applications and drivers who must develop custom extensions to the standard TIGA graphics library. Included in this package are the TMS340 Family Code Generation Tools, TIGA Software Developer's Package, TMS340 Family Graphics Library, and TMS340 C Source Debugger. The software runs on the IBM PC-AT and compatible MS-DOS machines. *TMS340 Family Code Generation Tools:* Includes a C compiler, TMS340 assembler, linker, archiver, and ROM utility. These software tools generate code for both the TMS34010 and TMS34020. TMS34082 support is provided. *TIGA Software Developer's Package:* Provides the utilities needed by the developer to produce TIGA drivers for software applications. *TMS340 Family Graphics Library:* Provides source and object code for the standard TIGA graphics functions. This package provides a convenient starting point for those developing TMS34010 and TMS34020 code for non-TIGA environments. This product replaces two older products: the TMS34010 Math/Graphics Function Library and the TMS34010 Font Library. *TMS340 Family C Source Debugger:* Provides symbolic and high-level language facilities for debugging code for the TMS34010 and TMS34020 written in both C and TMS340 assembly language. Runs on any TIGA board.
TMS340SPK-PC	**TIGA Software Porting Kit** The SPK meets the needs of OEM equipment makers who are porting the TIGA software interface to their TMS34010- and TMS34020-based hardware products. This package contains the complete source code necessary to build and run TIGA on a TMS340-based board. The SPK also contains the TIGA driver for MS Windows 3.0, and the complete TMS340 Software Developer's Kit (described above).
SPVZ071	**TIGA Driver Developer's Kit** The TIGA DDK meets the needs of those developers of TIGA-compatible software applications and drivers who do not need to develop custom extensions to the standard TIGA graphics library. Included in this package are the TIGA application interface libraries, example programs, and other TIGA-related files and information. All that is needed in addition to the DDK is an IBM-compatible PC, a TIGA board, and a C compiler for MS-DOS machines.
TMDS3442203018	**TMS340 Family C Compiler for VAX/VMS**
TMDS3442213018	**TMS340 Family C Compiler for VAX**
TMDS3442553008	**TMS340 Family C Compiler for SUN**
TMDSTDB10	**TMS34010 TIGA Development Board**
TMDS3469910000	**TMS34010 XDS/22 Realtime Hardware Emulator, U.S.**
TMDS3469981000	**TMS34010 XDS/22 Realtime Hardware Emulator, Europe**

reset

Reset puts the TMS34010 into a known initial state. It is entered when the input signal at the $\overline{\text{RESET}}$ pin is asserted low. $\overline{\text{RESET}}$ must remain active low for a minimum of 40 local clock (LCLK1 and LCLK2) periods to ensure that the TMS34010 has sufficient time to establish its initial internal state.

While $\overline{\text{RESET}}$ remains asserted, all outputs are in a known state, no DRAM-refresh cycles take place, and no screen-refresh cycles are performed.

At the low-to-high transition of the $\overline{\text{RESET}}$ signal, the state of the $\overline{\text{HCS}}$ input determines whether the TMS34010 will be halted or begin executing instructions. The TMS34010 may be in one of two modes, host-present or self-bootstrap mode.

1. Host-Present Mode

   If $\overline{\text{HCS}}$ is high at the end of reset, TMS34010 instruction execution is halted and remains halted until the host clears the HLT (halt) bit in HSTCTL (host control register). Following reset, the eight $\overline{\text{RAS}}$-only refresh cycles required to initialize the dynamic RAMs are performed automatically by the TMS34010 memory control logic. As soon as the eight $\overline{\text{RAS}}$-only cycles are completed, the host is allowed access to TMS34010 memory. At this time, the TMS34010 begins to automatically perform DRAM refresh cycles at regular intervals. The TMS34010 remains halted until the host clears the HLT bit. Only then does the GSP fetch the level-0 vector address from location 0FFFFFFE0h and begin executing its reset service routine.

2. Self-Bootstrap Mode

   If $\overline{\text{HCS}}$ is low at the end of reset, the TMS34010 first performs the eight $\overline{\text{RAS}}$-only refresh cycles required to initialize the DRAMs. Immediately following the eight $\overline{\text{RAS}}$-only cycles, the TMS34010 fetches the level-0 vector address from location 0FFFFFFE0h, and begins executing its reset service routine.

Unlike other interrupts and software traps, reset does not save previous ST or PC values. This is because the value of the stack pointer just before a reset is generally not valid, and saving its value on the stack is unnecessary. A TRAP 0 instruction, which uses the same vector address as reset, similarly does not save the ST or PC values.

asserting reset

A reset is initiated by asserting the $\overline{\text{RESET}}$ input pin at its active-low level. To reset the TMS34010 at power up, $\overline{\text{RESET}}$ must remain active low for a minimum of 40 local clock periods after power levels have become stable. At times other than power up, the TMS34010 is also reset by holding $\overline{\text{RESET}}$ low for a minimum of 40 clock periods. The 40-clock interval is required to bring TMS34010 internal circuitry to a known initial state. While $\overline{\text{RESET}}$ remains asserted, the output and bidirectional signals are driven to a known state.

The TMS34010 drives its $\overline{\text{RAS}}$ signal inactive high as long as $\overline{\text{RESET}}$ remains low. The specifications for certain DRAM and VRAM devices, including the TMS4161, TMS4164 and TMS4464 devices, require that the $\overline{\text{RAS}}$ signal be driven inactive-high for 100 microseconds during system reset. Holding $\overline{\text{RESET}}$ low for 150 microseconds will cause the $\overline{\text{RAS}}$ signal to remain high for the 100 microseconds required to bring the memory devices to their initial states. DRAMs such as the TMS4256 specify an initial $\overline{\text{RAS}}$ high time of 200 microseconds, requiring that $\overline{\text{RESET}}$ be held low for 250 microseconds. In general, holding $\overline{\text{RESET}}$ low for t microseconds ensures that $\overline{\text{RAS}}$ remains high initially for $t - 50$ microseconds.

suspension of DRAM-refresh cycles during reset

An active-low level at the $\overline{\text{RESET}}$ pin is considered to be a power-up condition, and DRAM refresh is not performed until $\overline{\text{RESET}}$ goes inactive high. Consequently, the previous contents of the local memory may not be valid after a reset.

initial state following reset

While the $\overline{\text{RESET}}$ pin is asserted low, the TMS34010's output and bidirectional pins are forced to the states listed below.

INITIAL STATE OF PINS FOLLOWING A RESET

OUTPUTS DRIVEN TO HIGH LEVEL	OUTPUTS DRIVEN TO LOW LEVEL	BIDIRECTIONAL PINS DRIVEN TO HIGH IMPEDANCE
DDOUT	$\overline{\text{BLANK}}$	$\overline{\text{HSYNC}}$
HRDY		$\overline{\text{VSYNC}}$
$\overline{\text{DEN}}$		HD0-HD15
$\overline{\text{LAL}}$		LAD0-LAD15
$\overline{\text{TR/QE}}$		
$\overline{\text{RAS}}$		
$\overline{\text{CAS}}$		
$\overline{\text{W}}$		
$\overline{\text{HINT}}$		
$\overline{\text{HLDA/EMUA}}$		

Immediately following reset, all I/O registers are cleared (set to 0h), with the possible exception of the HLT bit in the HSTCTL register. The HLT bit is set to 1 if $\overline{\text{HCS}}$ is high just prior to the low-to-high transition of $\overline{\text{RESET}}$.

Just prior to execution of the first instruction in the reset routine, the TMS34010's internal registers are in the following state:

- General-purpose register files A and B are uninitialized.
- The ST is set to 00000010h.
- The PC contains the 32-bit vector fetched from memory address 0FFFFFFE0h.

TMS34010 local memory interface

The TMS34010 local memory interface consists of a triple-multiplexed address/data bus on which row addresses, column addresses, and data are transmitted. The associated memory control signals support direct interfacing to both DRAMs and VRAMs. At the beginning of a typical memory cycle, the address is output in multiplexed fashion as a row address followed by a column address. The remainder of the cycle is used to transfer data between the TMS34010 and memory.

TMS34010 local memory interface

The TMS34010 local memory interface consists of a triple-multiplexed address/data bus on which row addresses, column addresses, and data are transmitted. The associated memory control signals support direct interfacing to both DRAMs and VRAMs. At the beginning of a typical memory cycle, the address is output in multiplexed fashion as a row address followed by a column address. The remainder of the cycle is used to transfer data between the TMS34010 and memory.

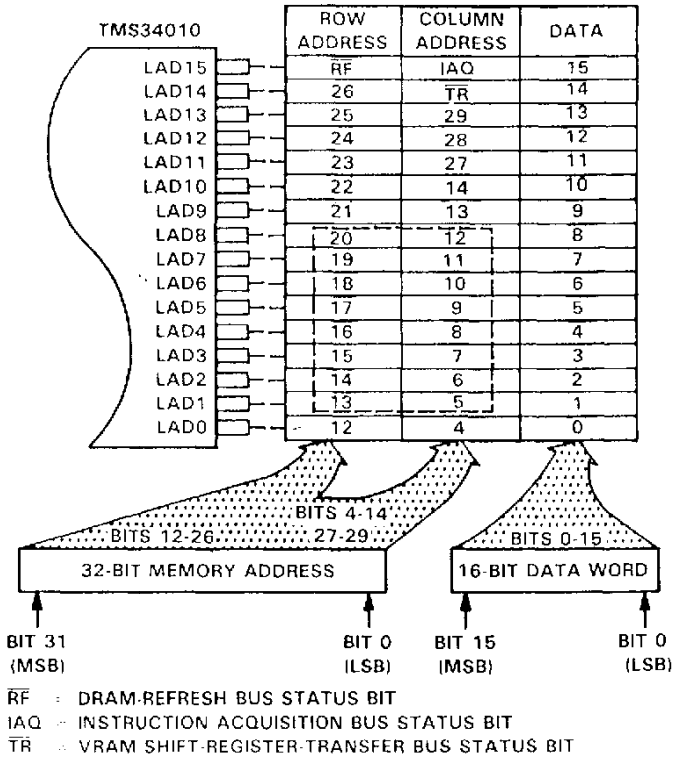


FIGURE 7. TRIPLE MULTIPLEXING OF ADDRESSES AND DATA

The following types of memory cycles are supported: read, write, VRAM memory-to-register, VRAM register-to-memory, $\overline{RAS}$-only DRAM refresh and $\overline{CAS}$-before-$\overline{RAS}$ DRAM refresh. The functional timing for these cycles is shown in the next six figures. Each memory cycle is a minimum of two maching states (a state is one local clock period) in duration. The seventh figure indicates the timing signals output during an internal cycle, i.e., a cycle during which no memory access takes place. An internal cycle is one state in duration.

During a memory cycle, the row address, column address, and data are transmitted over the same physical bus lines. The manner in which logical addresses are output at the memory interface makes external multiplexing hardware unnecessary, while supporting a wide variety of memory configurations. For example, in Figure 7, 16 consecutive address bits (5 through 20) are output on LAD1-LAD8 during the row and column address times. Output along with the address are bus status signals that indicate when DRAM refresh cycles, screen refresh (VRAM memory-to-register) cycles, and instruction fetch cycles are occurring.

The following remarks apply to memory timing in general. A row address is output on LAD0-LAD15 at the start of the cycle, and is valid before and after the fall of $\overline{RAS}$. Next a column address is output on LAD0-LAD15. The column address is valid briefly before and after the falling edge of $\overline{LAL}$, but is not valid at the falling edge of $\overline{CAS}$. The column address is clocked into an external transparent latch (e.g., a 74AS373 octal latch) on the falling edge of $\overline{LAL}$ to provide the hold time on the column address required for dynamic RAMs and video RAMs. A transparent latch is required in order that the row address be available at the outputs of the latch during the start of the cycle.

Very large memory configurations may require external buffering of data lines. The $\overline{DEN}$ signal serves as the drive-enable signal to external bidirectional buffers, e.g., 74AS245 octal buffers. The DDOUT signal serves as the direction control for the buffers.

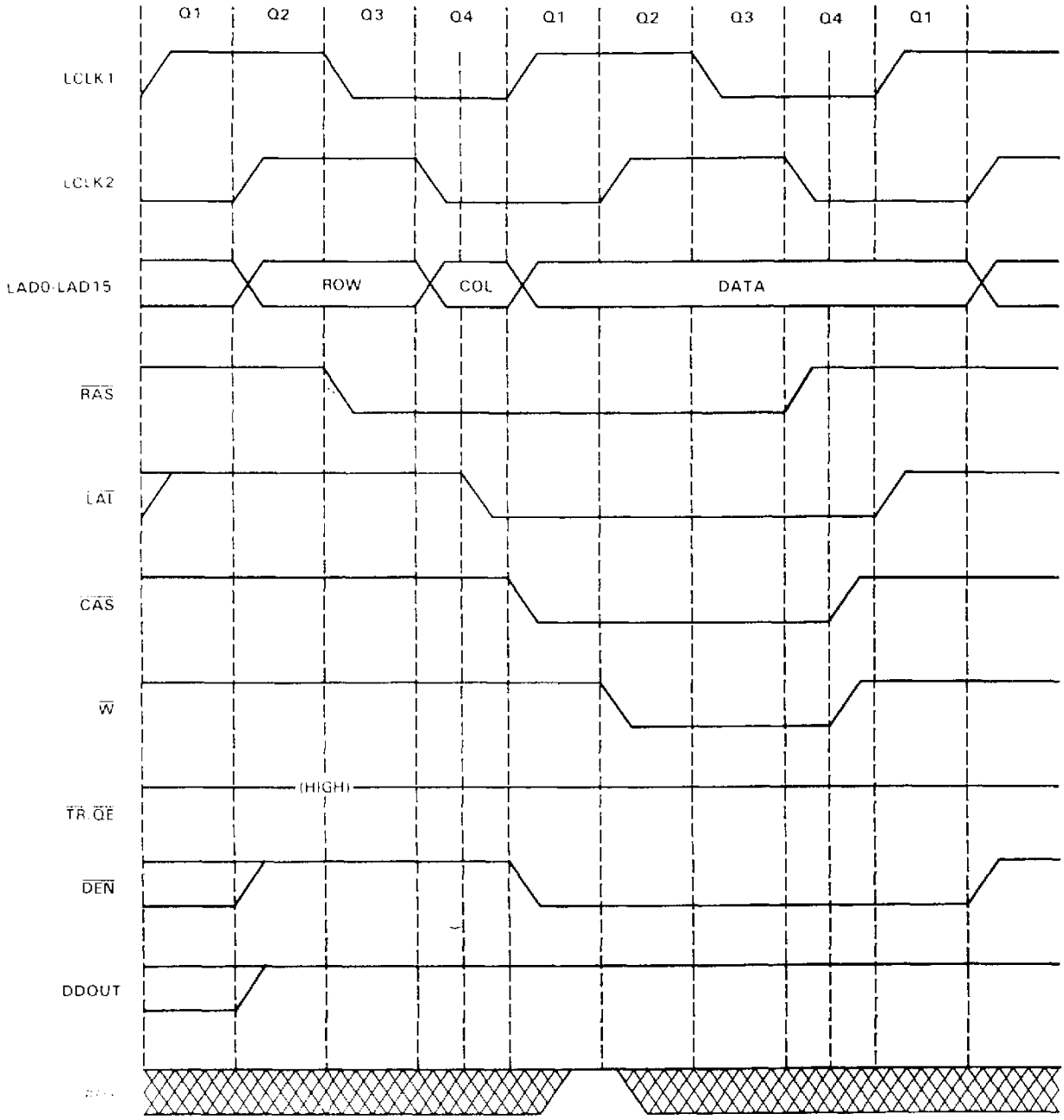
When an I/O register is addressed by the TMS34010, a special memory read or write cycle is performed. During this cycle, the external $\overline{RAS}$ signal falls, but the external $\overline{CAS}$ remains inactive-high for the duration of the cycle.

The timing shown in the first six functional timing diagrams assumes that the LRDY input remains high during the cycle. The LRDY pin is pulled low by slower memories requiring a longer cycle time. The TMS34010 samples the LRDY input at the end of Q1, as indicated in the figures. If LRDY is low, the TMS34010 inserts an additional state, called a "wait" state, into the cycle. Wait states continue to be inserted until LRDY is sampled at a high level. The cycle then completes in the manner indicated in the functional timing diagrams. A wait state is one local clock period in duration. Three additional timing diagrams provide examples of cycles extended by wait states.

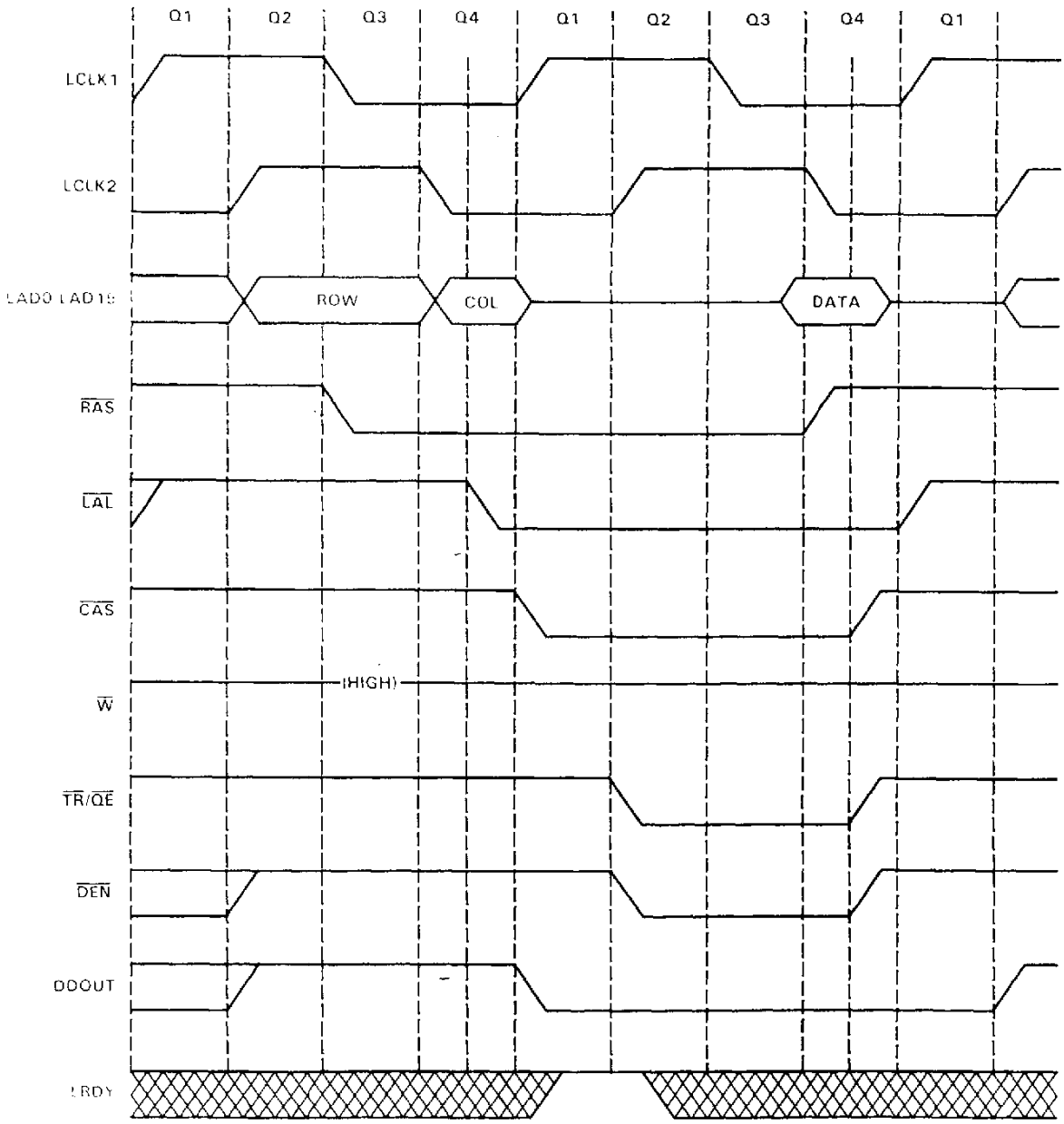
The LRDY input is ignored by the TMS34010 during internal cycles.

A hold/hold acknowledge capability is also built into the local memory interface to allow external devices to request control of the bus. After acknowledging a hold request, the TMS34010 releases the bus by driving its address/data bus and control outputs into high impedance.

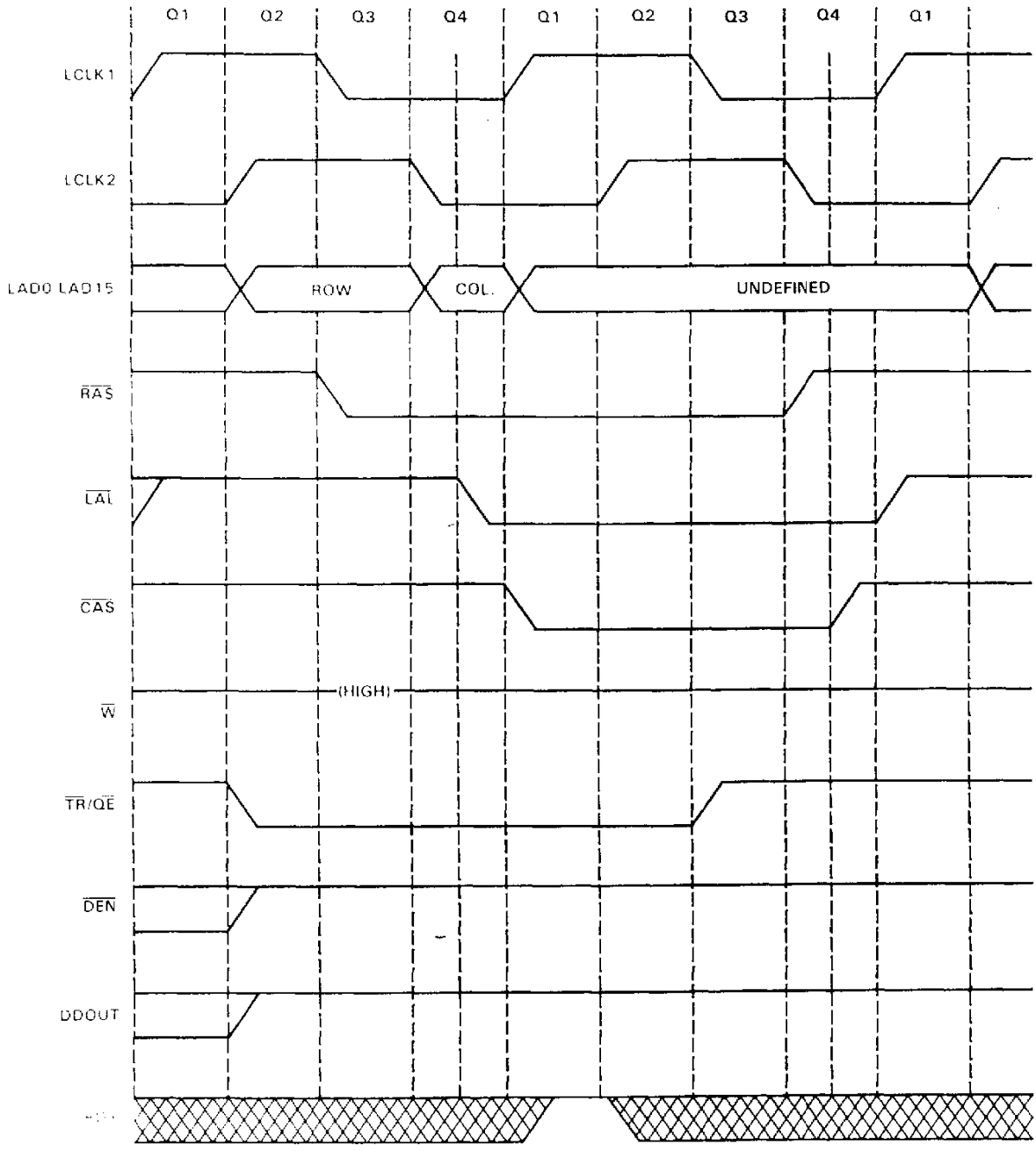
write cycle timing



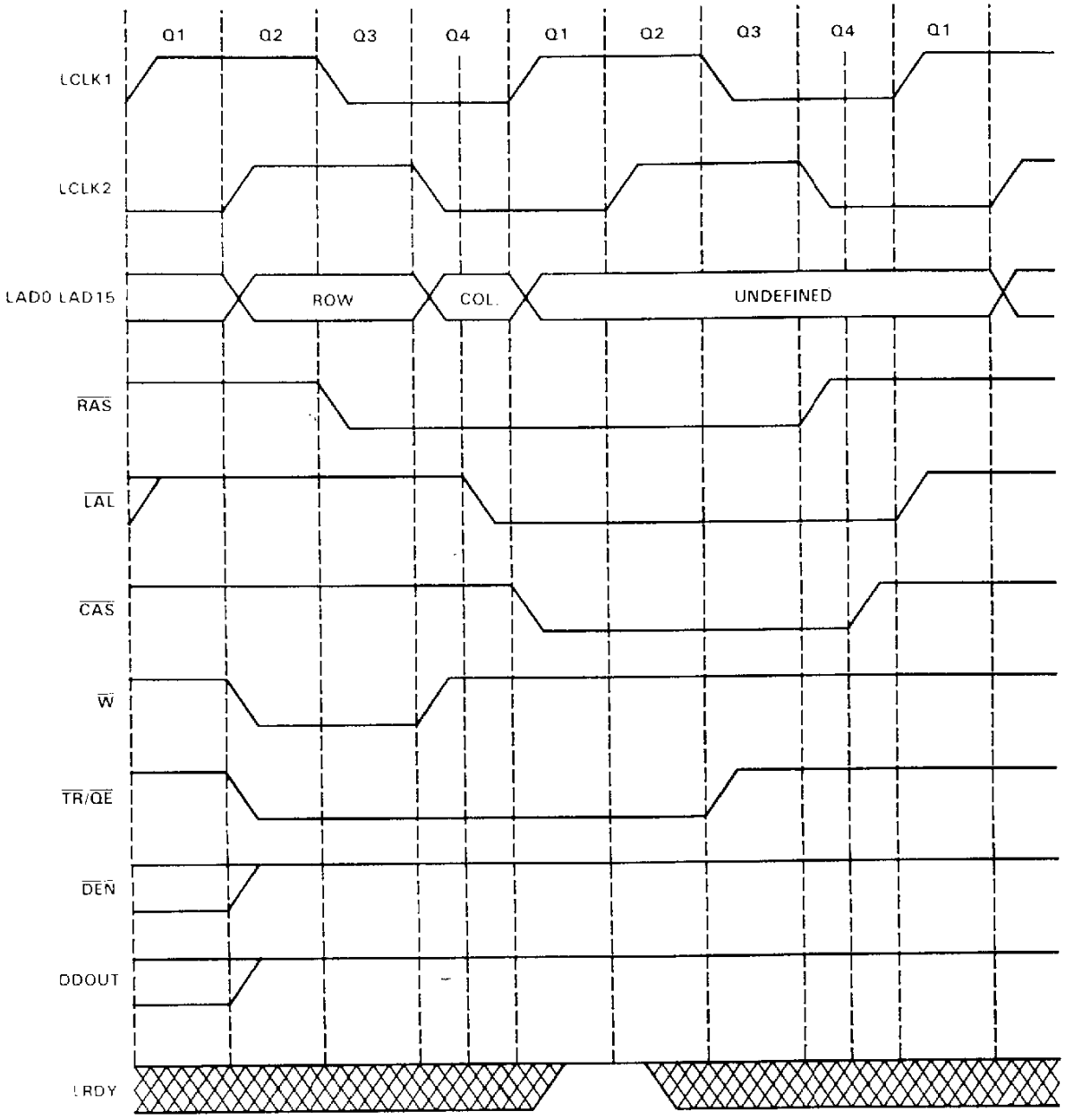
read cycle timing



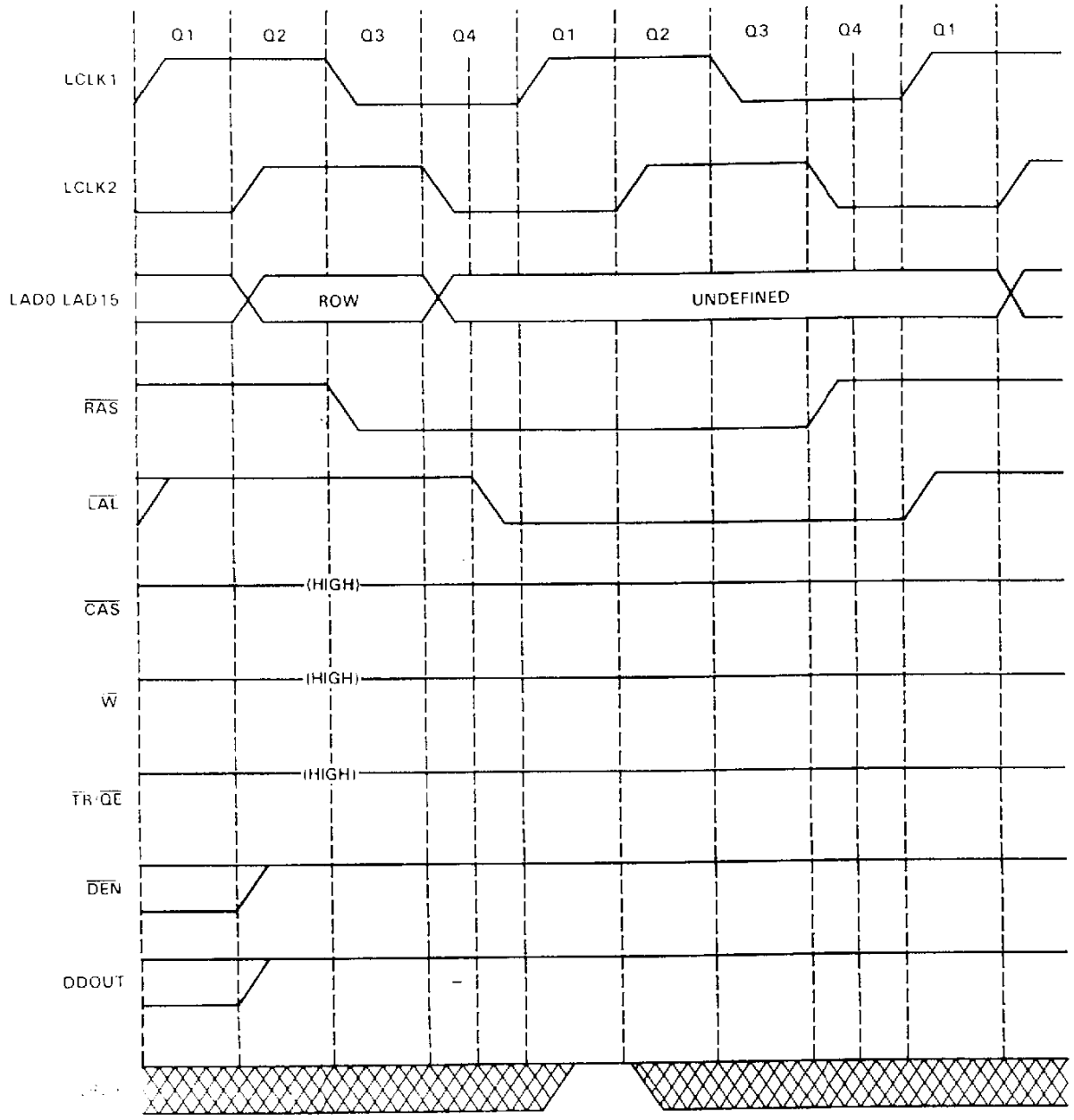
memory-to-register cycle timing



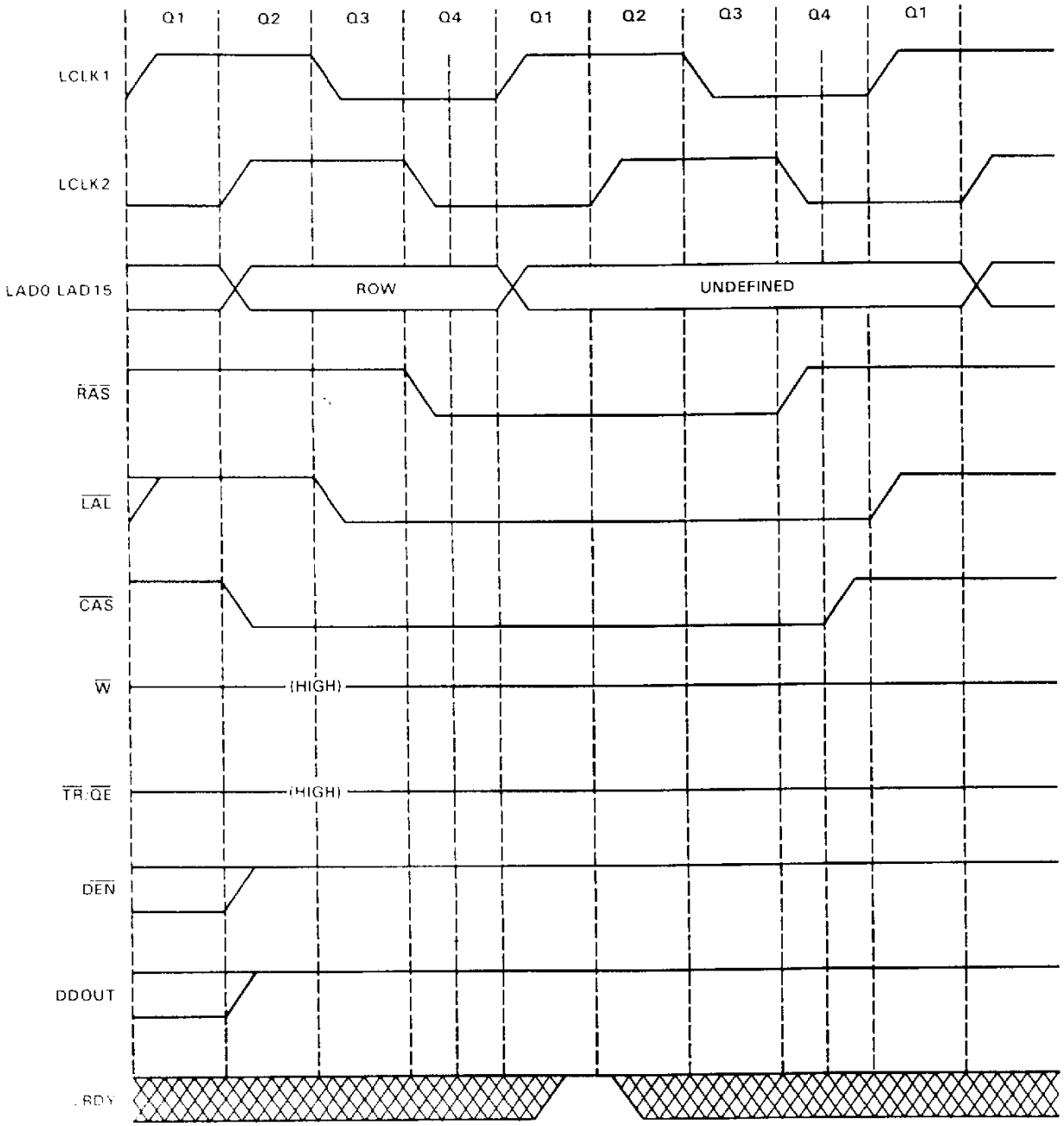
register-to-memory cycle timing



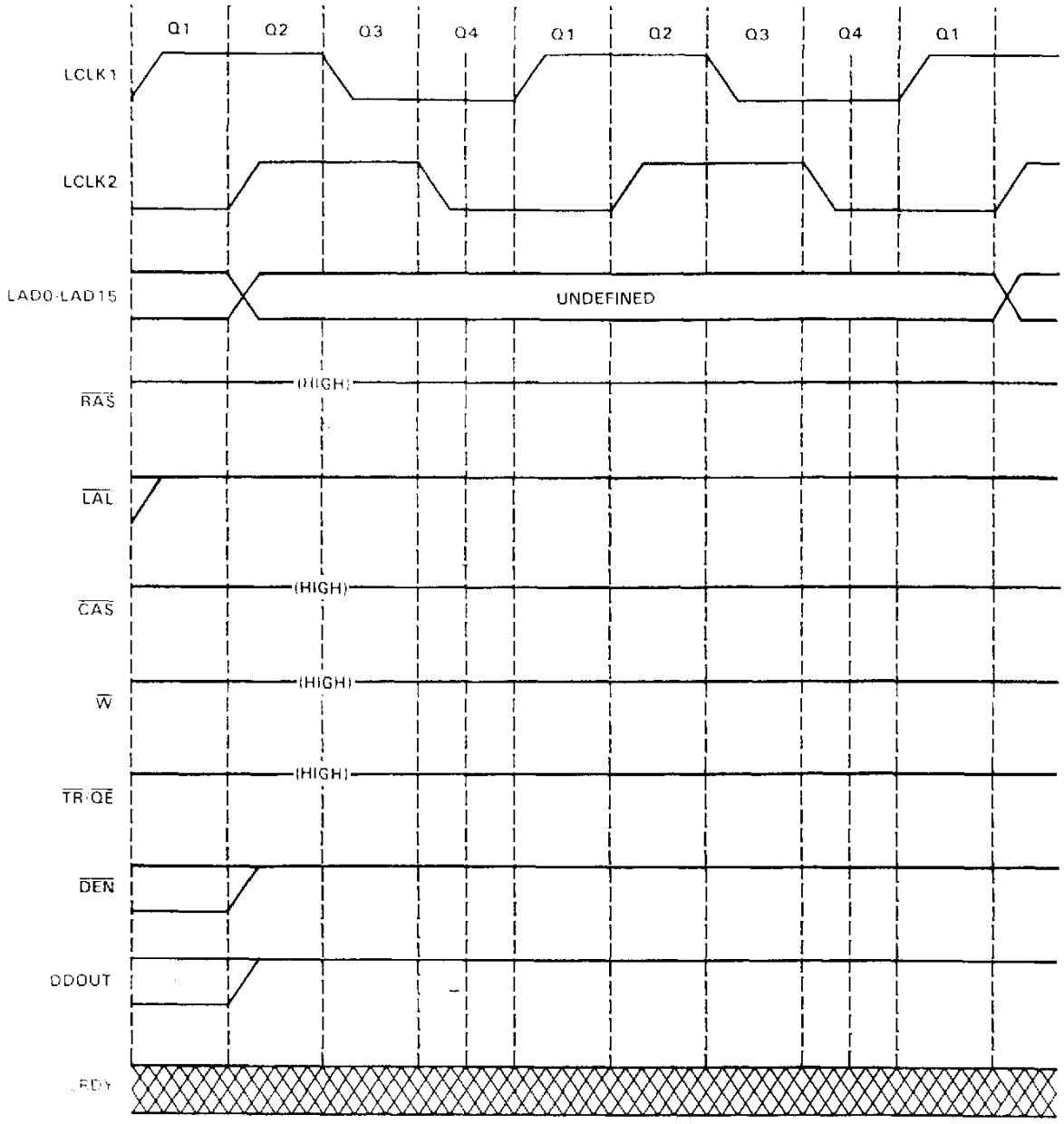
$\overline{RAS}$-only DRAM refresh cycle timing



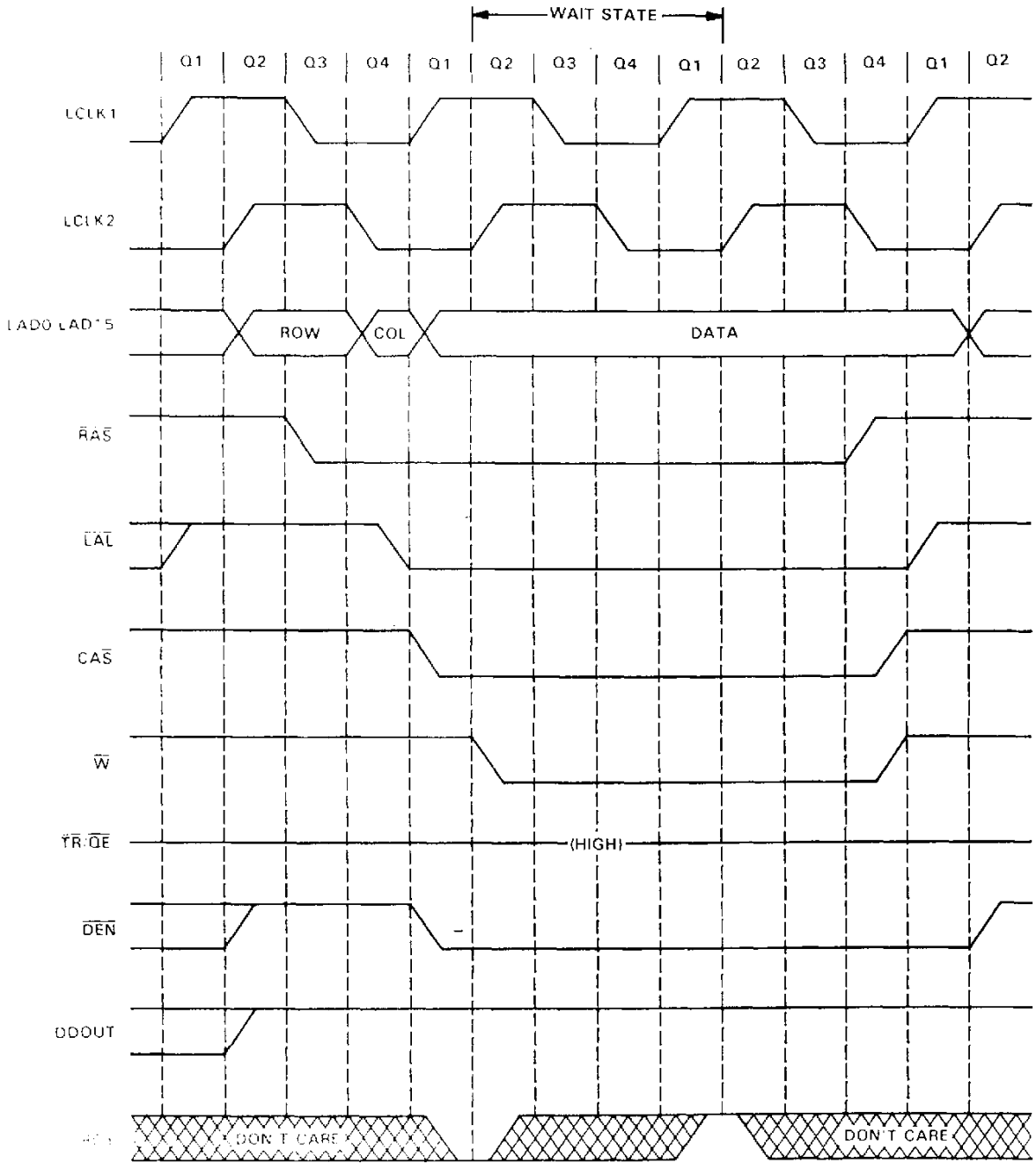
$\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh cycle timing



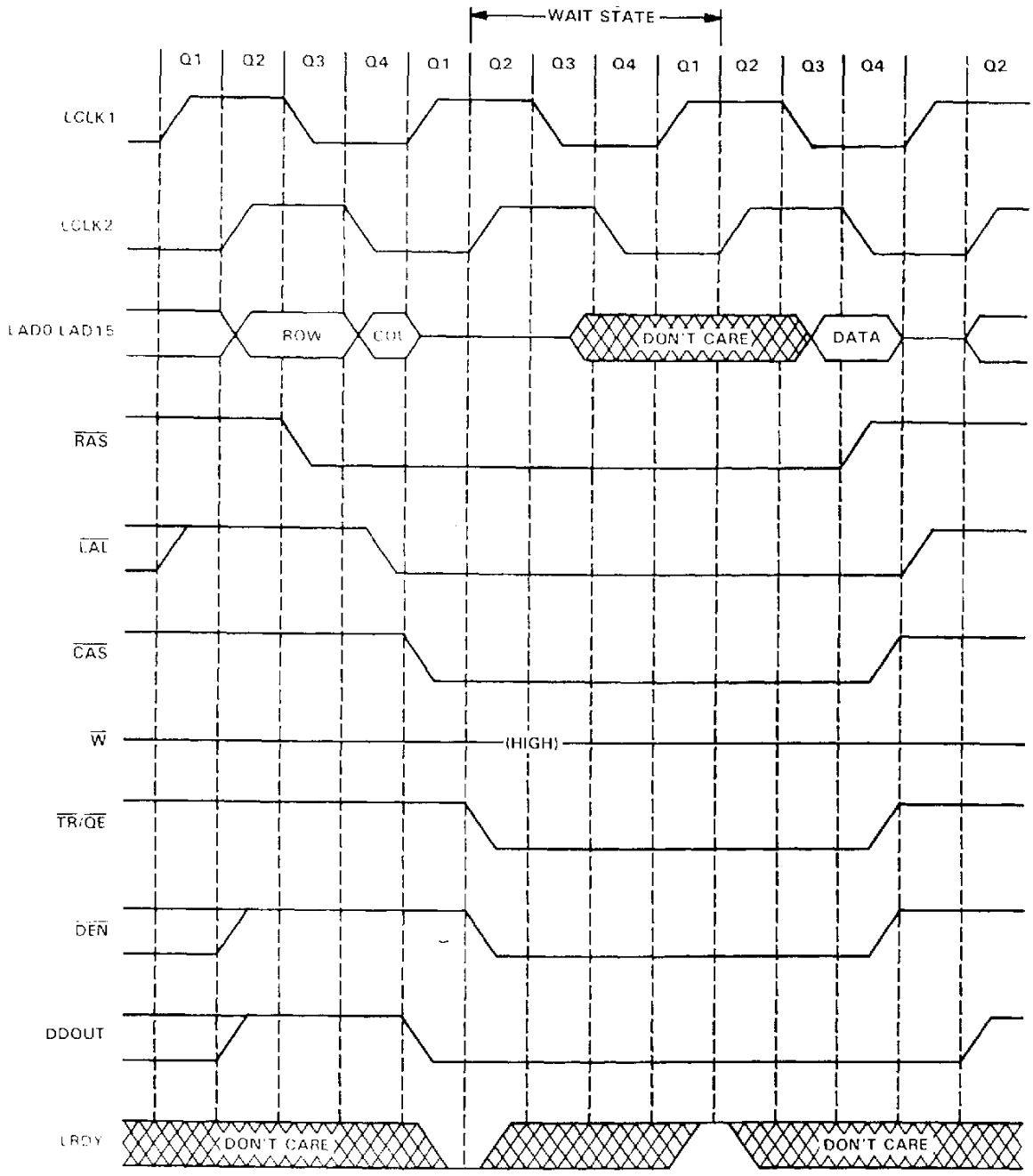
internal cycles back to back



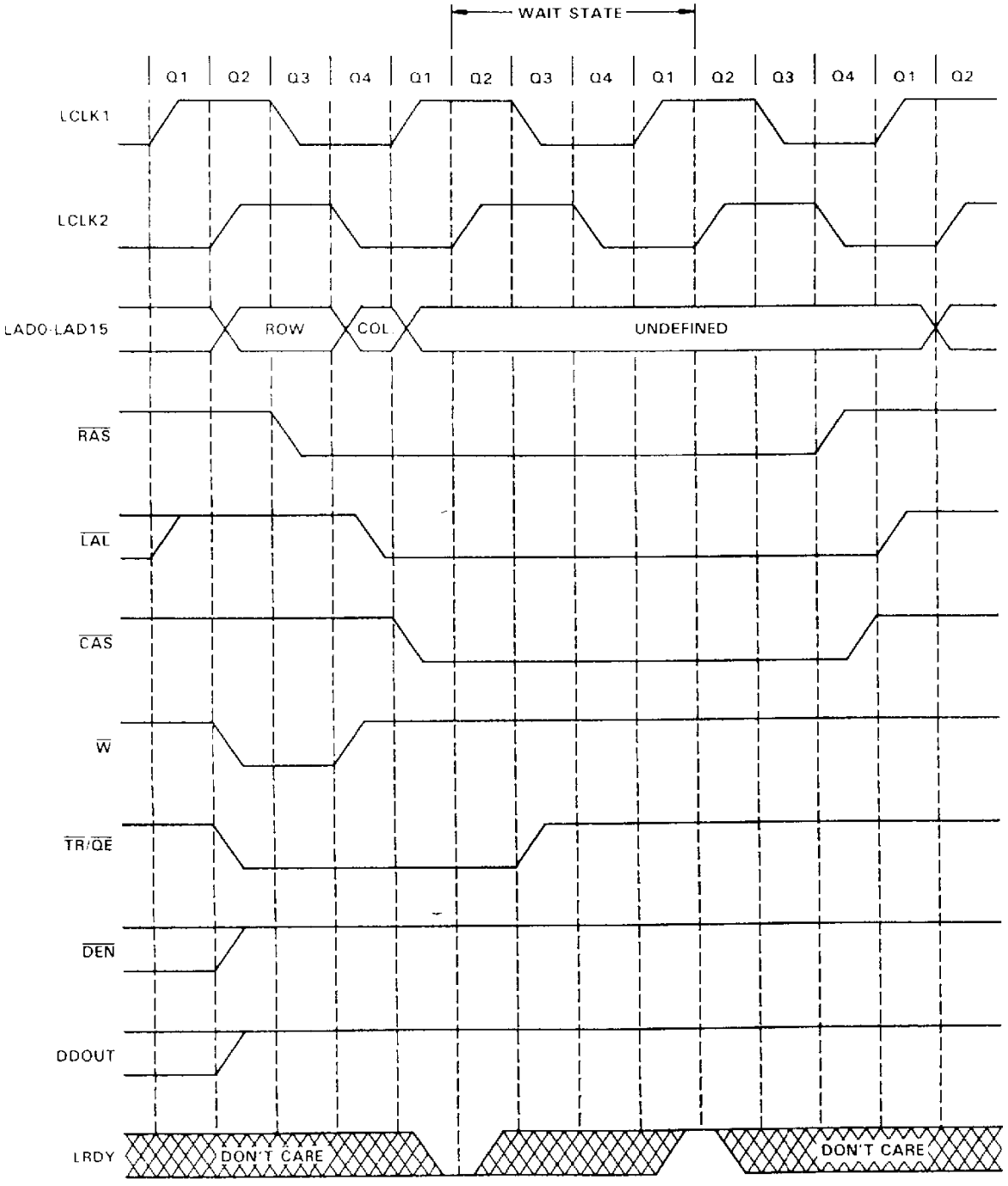
write cycle with one wait state



read cycle with one wait state



register-to-memory cycle with one wait state



absolute maximum ratings over operating free-air temperature range†

Supply voltage, V_{CC} . . . . . 7 V
Input voltage range . . . . . −0.3 V to 20 V
Off-state output voltage range . . . . . −2 V to 7 V
Operating free-air temperature range . . . . . 0°C to 70°C
Storage temperature range . . . . . −10°C to 150°C

†Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
All voltage values are with respect to the V_{SS} pins of the chip.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	4.75	5.0	5.25	V
V_{SS}	Supply voltage†	0	0	0	V
I_{OH}	High-level output current			−400	μA
I_{OL}	Low-level output current			2.0‡	mA
T_A	Operating free-air temperature	0		70	°C

†Care should be taken by card designers to provide a minimum inductance path between the V_{SS} pins and system ground in order to minimize V_{SS} noise.
‡Output current of 2.0 mA is sufficient to drive five low-power Schottky TTL loads or 10 advanced low-power Schottky TTL loads (worst case).

DC electrical characteristics

PARAMETER			TEST CONDITIONS		MIN†	TYP‡	MAX†	UNIT
V_{IH}§	High-level input voltage, TTL-level signal	All inputs except INCLK	V_{CC} = 5.0 V		2.2		V_{CC}+0.3	V
		INCLK			3.0		V_{CC}+0.3	
V_{IL}	Low-level input voltage, TTL-level signal	All inputs except INCLK			−0.3		0.8	V
		INCLK			−0.3		0.8	
V_{OH}	High-level output voltage, TTL-level signal		V_{CC} = min, I_{OH} = max,		2.6			V
V_{OL}	Low-level output voltage, TTL-level signal		V_{CC} = max, I_{OL} = max,				0.6	V
I_O	High impedance leakage current, bidirectional pins		V_{CC} = max	V_O = 2.8 V			20	μA
				V_O = 0.6 V			−20	
I_I	Input current	All inputs except RUN/$\overline{EMU}$§	V_I = V_{SS} to V_{CC}				±20	μA
I_{CC}	Supply current		V_{CC} = max, 40 MHz				125	mA
			V_{CC} = max, 50 MHz				150	
			V_{CC} = max, 60 MHz				175	
C_I	Input capacitance					10		pF
C_O	Output capacitance (except address/data lines)					10		pF

†For conditions shown as "min" or "max," use the appropriate value specified under "Recommended Operating Conditions."
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§RUN/$\overline{EMU}$ will be no connected in a typical configuration. The nominal pull-up current will be 250 μA.

signal transition levels

FIGURE 8. TTL-LEVEL OUTPUTS

TTL-level outputs are driven to a minimum logic-high level of 2.6 volts and to a maximum logic-low level of 0.6 volts. Output transition times are specified as follows.

For a high-to-low transition on a TTL-compatible output signal, the level at which the output is said to be "no longer high" is 2.0 volts, and the level at which the output is said to be "low" is 0.8 volts. For a low-to-high transition, the level at which the output is said to be "no longer low" is 0.8 volts, and the level at which the output is said to be "high" is 2.0 volts.

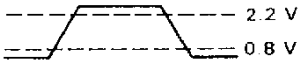
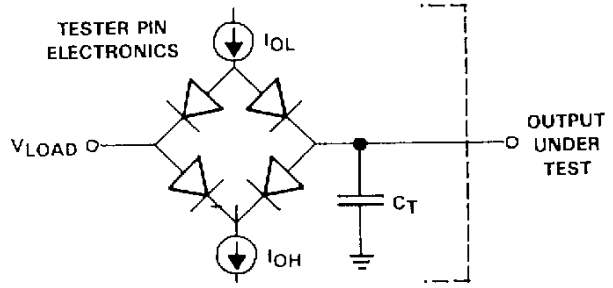


FIGURE 9. TTL-LEVEL INPUTS

Transition times for TTL-compatible inputs are specified as follows. For a high-to-low transition on an input signal, the level at which the input is said to be "no longer high" is 2.2 volts, and the level at which the input is said to be "low" is 0.8 volts. For a low-to-high transition on an input signal, the level at which the input is said to be "no longer low" is 0.8 volts, and the level at which the input is said to be "high" is 2.2 volts.

test measurement

The test load circuit shown in Figure 10 represents the programmable load of the tester pin electronics, which are used to verify timing parameters of TMS34010 output signals.



Where: I_{OL} = 2.0 mA DC level verification (all outputs)
I_{OH} = 400 μA (all outputs)
V_{LOAD} = 1.5 V DC level verification
0.7 V Timing verification
C_T = 65 pF typical load circuit capacitance

FIGURE 10. TEST LOAD CIRCUIT

timing parameter symbology

Timing parameter symbols have been created in accordance with JEDEC Standard 100. In order to shorten the symbols, some of the pin names and other related terminology have been abbreviated as follows:

AL	$\overline{\text{LAL}}$	HS	$\overline{\text{HSYNC}}$ or $\overline{\text{VSYNC}}$
C	$\overline{\text{CAS}}$	ICK	INCLK
CA	Column address	LR	$\overline{\text{LRDY}}$
CK	LCLK1 and LCLK2	QE	$\overline{\text{TR/QE}}$, when used as output enable
CK1	LCLK1	R	$\overline{\text{RAS}}$
CK2	LCLK2	RA	Row address
CS	$\overline{\text{HCS}}$	RS	$\overline{\text{HREAD}}$
D	Data	RY	$\overline{\text{HRDY}}$
DD	DDOUT	S	$\overline{\text{HREAD}}$ or $\overline{\text{HWRITE}}$
EN	$\overline{\text{DEN}}$	TR	$\overline{\text{TR/QE}}$, when used as shift register enable
F	HFS0, HFS1	VCK	VCLK
HK	$\overline{\text{HLDA}}$/$\overline{\text{EMUA}}$	W	$\overline{\text{W}}$
HR	$\overline{\text{HOLD}}$	WS	$\overline{\text{HWRITE}}$

Lowercase subscripts and their meaning are:

a	access time
c	cycle time (period)
d	delay time
h	hold time
su	setup time
t	transition time
w	pulse duration (width)

The following additional letters and symbols and their meaning are:

H	High
L	Low
V	Valid
Z	High impedance
↑	No longer low
↓	No longer high

host interface timing parameters

The timing parameters for host interface signals are shown in the next four figures. The purpose of these figures and the accompanying table is to quantify the timing relationships among the various signals. The explanation of the logical relationships among signals will be found in the *TMS34010 User's Guide* (number SPVU001B).

The *write strobe* referred to in the following table is the enabling signal during a write to one of the host interface registers (see comment 2 on the next page). Similarly, the *read strobe* is the enabling signal during a read.

Quarter clock time t_Q, which appears in the following table, is one quarter of a local output clock period, or twice the input clock period, $t_{c(ICK)}$.

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{su(FV-SL)}$	Setup time of $\overline{HWRITE}/HREAD$ high or HFS0, HFS1 valid to read or write strobe↓	10		10		ns
2	$t_{d(WSL-DV)}$	Delay from write strobe ↓ to data in valid, write cycle		$2t_Q$		$2t_Q$	ns
3	$t_{d(SL-SL)}$	Delay from read or write strobe low to next read or write strobe↓	$7t_Q+10$		$7t_Q+10$		ns
4	$t_{w(SL)}$	Duration of read or write strobe low	80		80		ns
5	$t_{d(SH-SL)}$	Delay from read or write strobe high to next read or write strobe↓	60		60		ns
6	$t_{h(WSH-DV)}$	Hold time of data in valid after write strobe high, write cycle	2		2		ns
7	$t_{h(SH-FV)}$	Hold time of $\overline{HWRITE}/HREAD$ high or HFS0, HFS1 valid after read or write strobe high	10		10		ns
8	$t_{h(RSL-DZ)}$	Hold time of data high impedance after read strobe↓, read cycle	0§		0§		ns
9	$t_{d(RSL-DV)}$	Delay from read strobe low to data out valid, read cycle with no wait		90		90	ns
10	$t_{h(RSH-DV)}$	Hold time of data out valid after read strobe↑, read cycle	0		0		ns
11	$t_{d(RSH-DZ)}$	Delay from read strobe high to data out high impedance, read cycle		30§		30§	ns
12	$t_{h(CSL-RYH)}$	Hold time of HRDY high after $\overline{HCS}$↓, cycle with wait	0		0		ns
13	$t_{d(CSL-RYL)}$	Delay from $\overline{HCS}$ low to HRDY low, cycle with wait		40		40	ns
14	$t_{w(RYL)}$	Pulse duration of HRDY low, cycle with wait		†		†	ns
15	$t_{d(RYL-RYH)}$	Delay from HRDY↓ to HRDY high, cycle with wait	0‡		0‡		ns
16	$t_{h(RYH-WSL)}$	Hold time of write strobe low after HRDY↑, write cycle with wait	40		40		ns
17	$t_{d(RYH-DV)}$	Delay from HRDY↑ to data out valid, read cycle with wait		30		30	ns
18	$t_{h(RYH-RSL)}$	Hold time of read strobe low after HRDY↑, read cycle with wait	40		40		ns

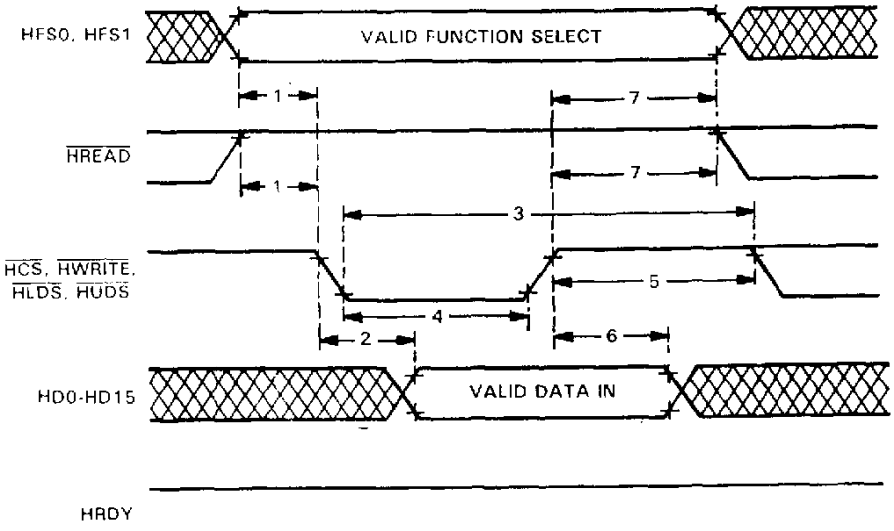
*Parameter 14 is a function of local bus memory contention. This parameter is not tested. Refer to the *TMS34010 User's Guide* for details.
‡Parameter 15 is specified as minimum 0 ns to indicate that a low-going pulse on HRDY can be arbitrarily narrow.
§These values are derived from characterization and are not tested.

general comments on host interface timing

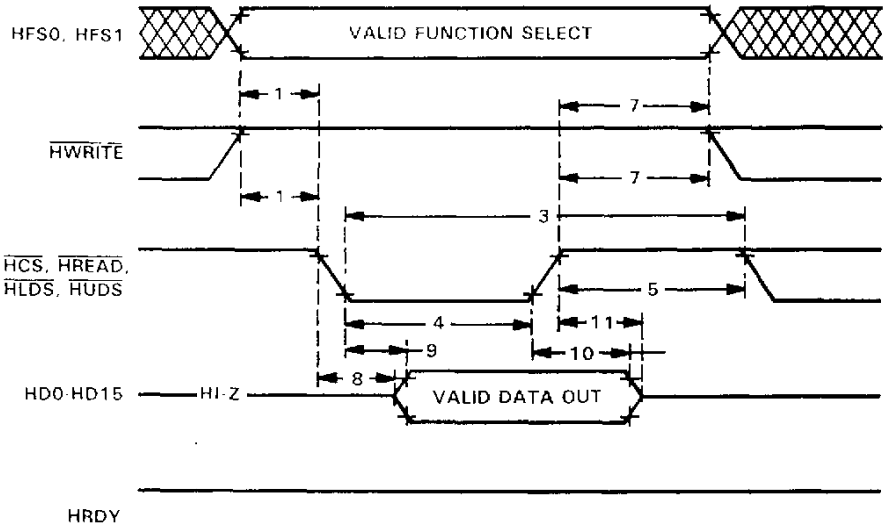
The following general comments apply to host interface timing:

1. The HRDY signal is enabled by an active low level on the $\overline{\text{HCS}}$ input. When $\overline{\text{HCS}}$ is inactive-high, HRDY is forced high regardless of the internal state of the device. Low-going transient pulses on $\overline{\text{HCS}}$ may result in low-going transient pulses on HRDY, but otherwise have no effect unless accompanied by active levels on other control signals.
2. A host interface write cycle occurs when $\overline{\text{HCS}}$, $\overline{\text{HWRITE}}$, and $\overline{\text{HLDS}}$ are low, or when $\overline{\text{HCS}}$, $\overline{\text{HWRITE}}$, and $\overline{\text{HUDS}}$ are low. The combination of these signals defines a *write strobe*. In either case, the last of the three signals to make the high-to-low transition is the strobe (write strobe) that begins the cycle. The first of the three signals to make the low-to-high transition ends the cycle. Similarly, a host interface read cycle occurs when $\overline{\text{HCS}}$, $\overline{\text{HREAD}}$, and $\overline{\text{HLDS}}$ are low, or when $\overline{\text{HCS}}$, $\overline{\text{HREAD}}$, and $\overline{\text{HUDS}}$ are low. The combination at these signals define a *read strobe*. In either case, the last of the three signals to make the high-to-low transition is the strobe (read strobe) that begins the cycle. The first of the three signals to make the low-to-high transition ends the cycle. All access times are specified with respect to the strobing edges that begin and end the cycle.
3. During a host interface read or write, $\overline{\text{HWRITE}}$ and $\overline{\text{HREAD}}$ must not be active-low simultaneously.
4. Host interface input signals $\overline{\text{HCS}}$, $\overline{\text{HUDS}}$, $\overline{\text{HLDS}}$, HFS0, HFS1, $\overline{\text{HREAD}}$, and $\overline{\text{HWRITE}}$ are assumed to be asynchronous with respect to the output clocks LCLK1 and LCLK2.

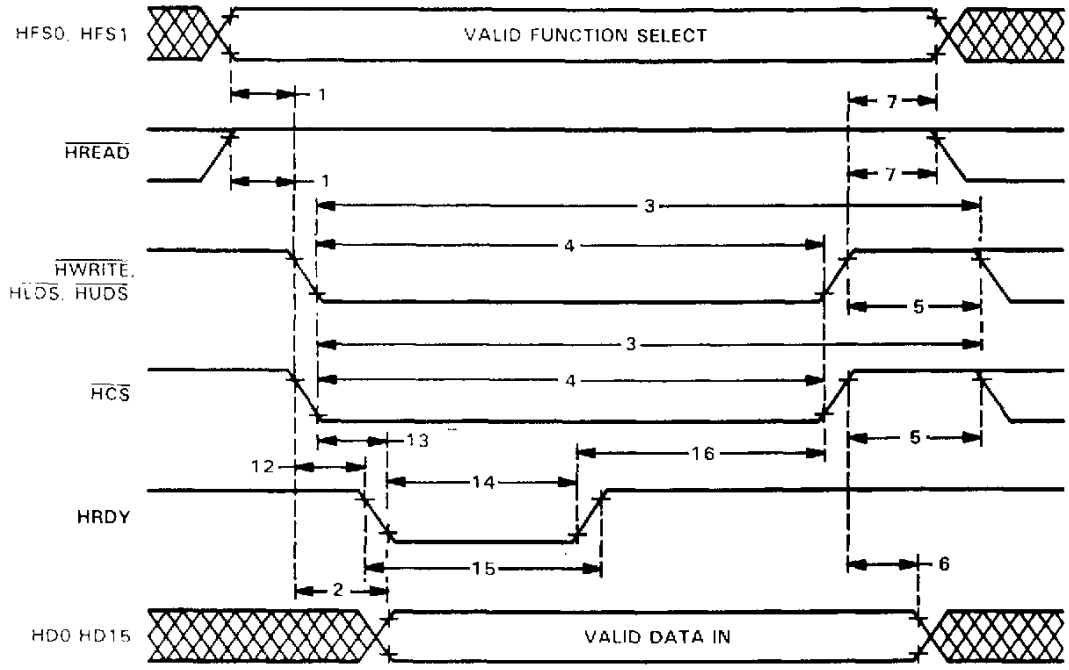
host interface timing: write cycle with no wait



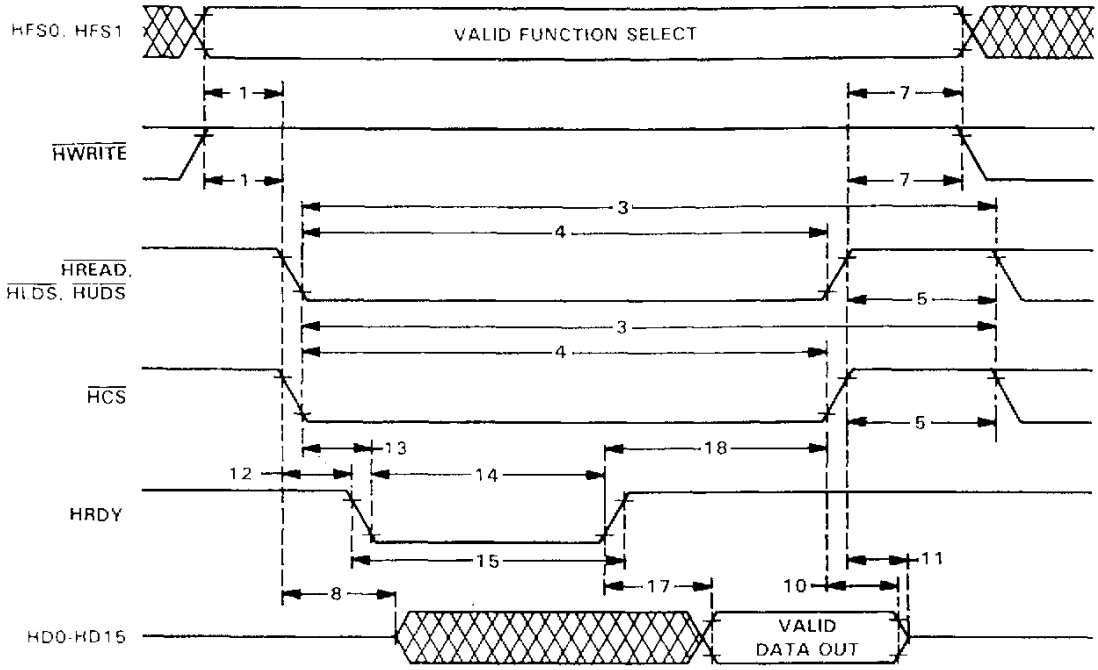
host interface timing: read cycle with no wait



host interface timing: write cycle with wait



host interface timing: read cycle with wait



reset timing

The timing parameters for device reset are shown in the next two figures. The purpose of these figures is to quantify the timing relationships among the $\overline{\text{RESET}}$, $\overline{\text{HCS}}$, and LCLK1 signals. $\overline{\text{RESET}}$ and $\overline{\text{HCS}}$ are asynchronous inputs that are internally synchronized by latches internal to the TMS34010. The timing relationships specified for these signals relative to LCLK1 need be met only to guarantee recognition of a transition of one of these signals at a particular clock edge. The explanation of the logical relationships among signals will be found in the *TMS34010 User's Guide*.

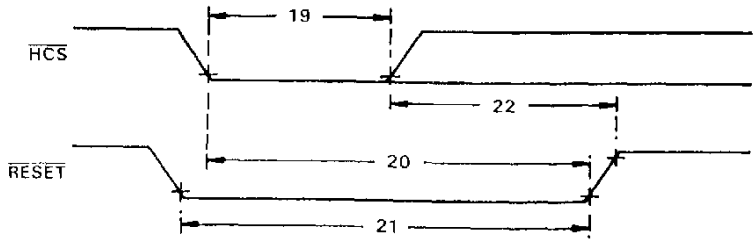
Quarter clock time t_Q which appears in the following table, is one quarter of a local output clock period, or twice the input clock period, $t_{c(ICK)}$.

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
19	$t_{w(CSL)}$	Duration of $\overline{\text{HCS}}$ low to configure GSP to run in self-bootstrap mode	$4t_Q+55$		$4t_Q+55$		ns
20	$t_{su(CSL-REH)}$	Setup time of $\overline{\text{HCS}}$ low to $\overline{\text{RESET}}\uparrow$ to configure the GSP to run in self-bootstrap mode	$8t_Q+55$		$8t_Q+55$		ns
21	$t_{w(REL)}$	Duration of $\overline{\text{RESET}}$ low to ensure that GSP is properly reset	$160t_Q-40$		$160t_Q-40$		ns
22	$t_{d(CSH-REH)}$	Delay from $\overline{\text{HCS}}\uparrow$ to $\overline{\text{RESET}}$ high, end of reset, to configure GSP to run in self-bootstrap mode		$4t_Q-50$†		$4t_Q-50$†	ns
23	$t_{su(REV-CK1L)}$	Setup time of $\overline{\text{RESET}}$ valid to LCLK1↓ to guarantee recognition at a particular clock edge	40‡		40‡		ns
24	$t_{h(CK1L-REV)}$	Hold time of $\overline{\text{RESET}}$ valid after LCLK1 low to guarantee recognition at a particular clock edge	10‡		10‡		ns
25	$t_{su(CSV-CK1L)}$	Setup time of $\overline{\text{HCS}}$ valid to LCLK1↓ to guarantee recognition at a particular clock edge	40‡		40‡		ns
26	$t_{h(CK1L-CSV)}$	Hold time of $\overline{\text{HCS}}$ valid after LCLK1 low to guarantee recognition at a particular clock edge	10‡		10‡		ns

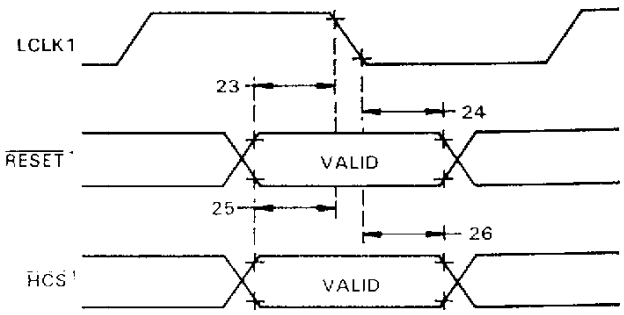
†Parameter 22 is the maximum amount by which the $\overline{\text{RESET}}$ low-to-high transition can be delayed after the $\overline{\text{HCS}}$ low-to-high transition and still guarantee that the GSP is configured to run in self-bootstrap mode (HLT bit = 0) following the end of reset. $\overline{\text{HCS}}$ may be held low for some time past the low-to-high $\overline{\text{RESET}}$ transition, and will be ignored by the GSP for 17 local clock periods following the clock edge at which the low-to-high $\overline{\text{RESET}}$ transition is detected. Following completion of the eight $\overline{\text{RAS}}$-only cycles that automatically follow reset, however, a low $\overline{\text{HCS}}$ level will be interpreted as a chip select.

‡$\overline{\text{RESET}}$ and $\overline{\text{HCS}}$ are asynchronous inputs. The specified setup and hold times of these signals with respect to the high-to-low transition of LCLK1 need be met only to guarantee that a transition of $\overline{\text{RESET}}$ or $\overline{\text{HCS}}$ is detected by the device at a particular clock edge.

reset: asynchronous timing relationships



reset: synchronous timing relationships



†RESET and HCS are asynchronous inputs. The specified setup and hold times of RESET or HCS with respect to the high-to-low LCLK1 transition must be met only to guarantee that a RESET or HCS transition is detected by the device at a particular clock edge.

local bus timing parameters

The following six figures show the timing parameters for the signals of the local memory interface bus, often simply referred to as the local bus. The purpose of these figures and the accompanying tables is to quantify the timing relationships among the various signals. The explanation of the logical relationships among signals will be found in the *TMS34010 User's Guide* (number SPVU001B).

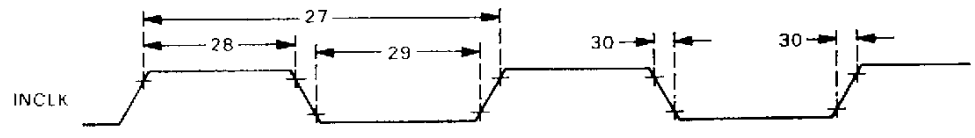
A number of parameter values are expressed in terms of quarter clock time t_Q, which is one quarter of a local clock period, or twice the input clock period, $t_{c(ICK)}$.

Input clock INCLK is divided internally by 8 to produce output clocks LCLK1 and LCLK2. Transitions of the other local interface output signals are also generated as delays from INCLK transitions. The divide-down logic that converts INCLK to the internal clocks used to generate LCLK1 and LCLK2 introduces significant propagation delays from the transitions of INCLK to the corresponding transitions of LCLK1 and LCLK2. While the frequency of INCLK is precisely eight times the frequency of LCLK1 or LCLK2, no timing relationship other than the frequency is specified between transitions of input clock INCLK and transitions of the output clocks LCLK1 and LCLK2.

NO	PARAMETER		TMS34010-40		TMS34010-50		TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
27	$t_{c(ICK)}$	Period of INCLK	25	62.5	20	62.5	16.5	62.5	ns
28	$t_{w(ICKH)}$	Pulse duration of INCLK high	8†		8†		6.5‡		ns
29	$t_{w(ICKL)}$	Pulse duration of INCLK low	8†		8†		6.5‡		ns
30	$t_{t(ICK)}$	Transition time (rise and fall) of INCLK	2†	8†	2†	8†	2†	8†	ns

†These values are based on computer simulation and are not tested.
‡This pulse width is tested at 1.4 volts.

local bus timing: input clock



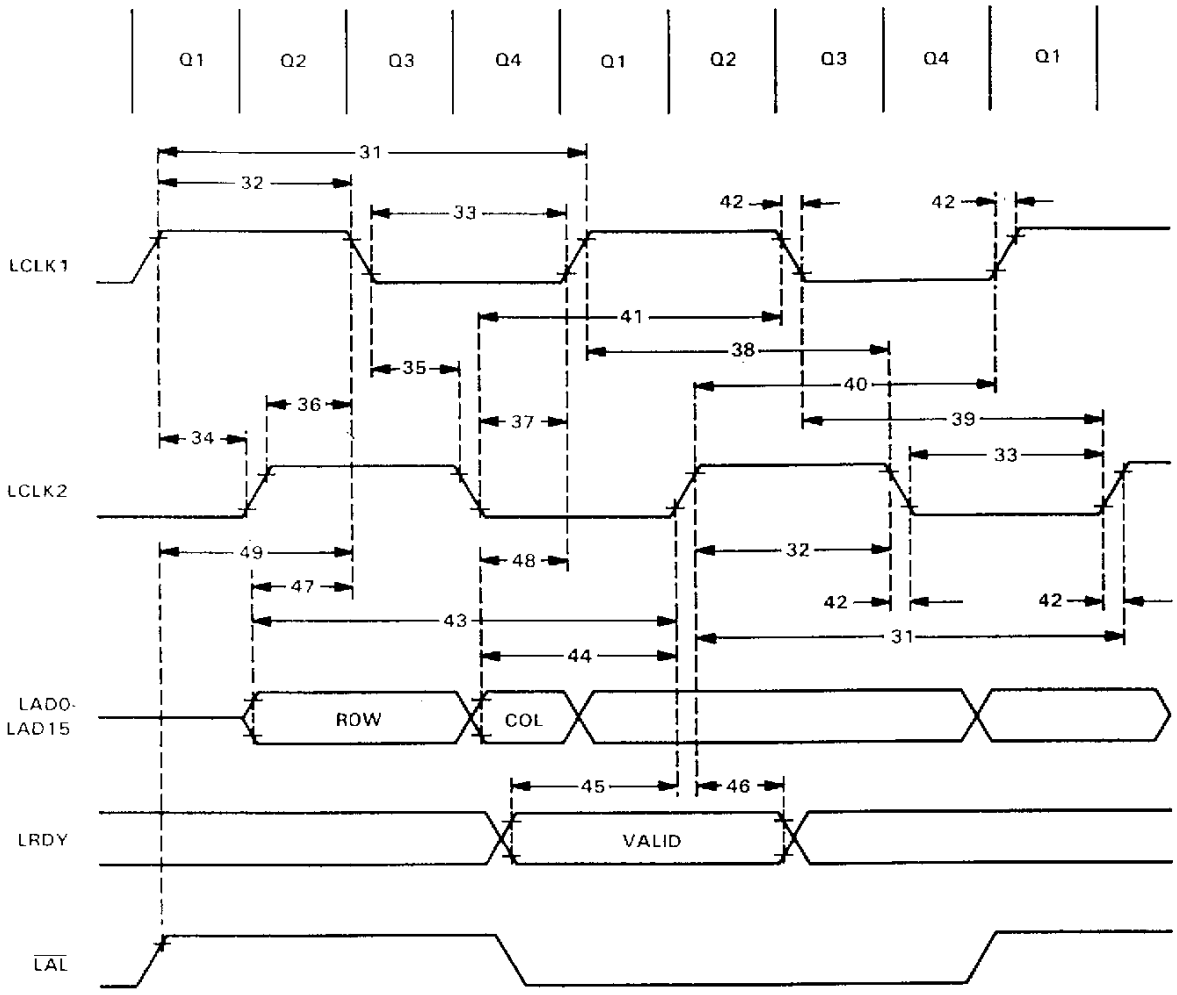
local bus timing parameters (continued)

Quarter clock time t_Q, which appears in the following table, is one quarter of a local output clock period, or $2t_{c(ICK)}$.

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
31	$t_{c(CK)}$	Period of local clocks LCLK1 and LCLK2	$8t_{c(ICK)}$†		$8t_{c(ICK)}$†		ns
32	$t_{w(CKH)}$	Pulse duration of local clock high	$2t_Q - 10$		$2t_Q - 10$		ns
33	$t_{w(CKL)}$	Pulse duration of local clock low	$2t_Q - 10$		$2t_Q - 10$		ns
34	$t_{h(CK1H-CK2L)}$	Hold time of LCLK2 low after LCLK1 high	$t_Q - 10$		$t_Q - 10$		ns
35	$t_{h(CK1L-CK2H)}$	Hold time of LCLK2 high after LCLK1 low	$t_Q - 10$		$t_Q - 10$		ns
36	$t_{h(CK2H-CK1H)}$	Hold time of LCLK1 high after LCLK2 high	$t_Q - 10$		$t_Q - 10$		ns
37	$t_{h(CK2L-CK1L)}$	Hold time of LCLK1 low after LCLK2 low	$t_Q - 10$		$t_Q - 10$		ns
38	$t_{h(CK1H-CK2H)}$	Hold time of LCLK2 high after LCLK1 high	$3t_Q - 10$		$3t_Q - 10$		ns
39	$t_{h(CK1L-CK2L)}$	Hold time of LCLK2 low after LCLK1 low	$3t_Q - 10$		$3t_Q - 10$		ns
40	$t_{h(CK2H-CK1L)}$	Hold time of LCLK1 low after LCLK2 high	$3t_Q - 10$		$3t_Q - 10$		ns
41	$t_{h(CK2L-CK1H)}$	Hold time of LCLK1 high after LCLK2 low	$3t_Q - 10$		$3t_Q - 10$		ns
42	t_t	Transition time (rise and fall) of LCLK1 or LCLK2		10		10	ns
43	$t_{su(RAV-CK2H)}$	Setup time of row address valid to LCLK2↑	$4t_Q - 25$		$4t_Q - 15$		ns
44	$t_{su(CAV-CK2H)}$	Setup time of column address valid to LCLK2↑	$2t_Q - 25$		$2t_Q - 15$		ns
45	$t_{su(LRV-CK2H)}$	Setup time of LRDY valid to LCLK2↑	30‡		30‡		ns
46	$t_{h(CK2H-LRV)}$	Hold time of LRDY valid after LCLK2 high	0‡		0‡		ns
47	$t_{su(RAV-CK1L)}$	Setup time of row address valid to LCLK1↓	$t_Q - 25$		$t_Q - 15$		ns
48	$t_{su(CAV-CK1H)}$	Setup time of column address valid to LCLK1↑	$t_Q - 25$		$t_Q - 15$		ns
49	$t_{su(ALH-CK1L)}$	Setup time of $\overline{CAL}$ high to LCLK1↓	$2t_Q - 20$		$2t_Q - 10$		ns

† This is a functional minimum and is not tested. This parameter can also be specified as $4t_Q$.
‡ LRDY is a synchronous input sampled during the low-to-high transition of LCLK2. The specified setup and hold times must be met for the device to operate properly.

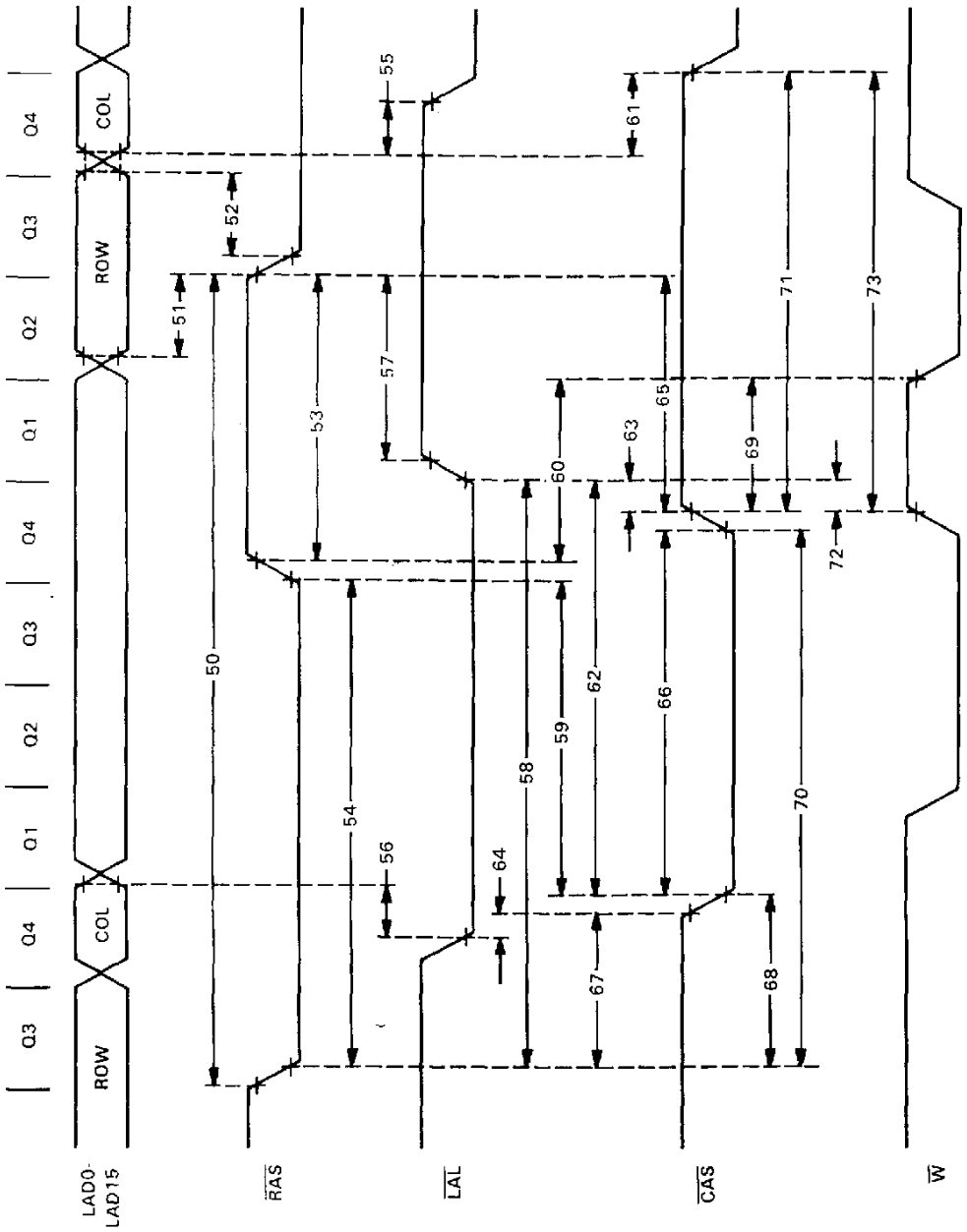
local bus timing: output clock and LRDY signal



NO.	PARAMETER		TMS34010-40		TMS34010-50		TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
50	$t_{d(RL\text{-}RL)}$	Delay from $\overline{RAS}\downarrow$ to $\overline{RAS}\downarrow$	$8t_Q$†		$8t_Q$†		$8t_Q$†		ns
51	$t_{su(RAV\text{-}RL)}$	Setup time of row address valid to $\overline{RAS}\downarrow$	t_Q-20		t_Q-15		t_Q-15		ns
52	$t_{h(RL\text{-}RAV)}$	Hold time of row address valid after $\overline{RAS}$ low	t_Q-20		t_Q-10		t_Q-5		ns
53	$t_{w(RH)}$	Pulse duration, $\overline{RAS}$ high	$3t_Q-20$		$3t_Q-10$		$3t_Q-5$		ns
54	$t_{w(RL)}$	Pulse duration, $\overline{RAS}$ low	$5t_Q-20$		$5t_Q-10$		$5t_Q-10$		ns
55	$t_{su(CAV\text{-}ALL)}$	Setup time of column address valid to $\overline{LAL}\downarrow$	$0.5t_Q-20$		$0.5t_Q-10$		$0.5t_Q-10$		ns
56	$t_{h(ALL\text{-}CAV)}$	Hold time of column address valid after $\overline{LAL}$ low	$0.5t_Q-15$		$0.5t_Q-10$		$0.5t_Q-10$		ns
57	$t_{h(ALH\text{-}RH)}$	Hold time of $\overline{RAS}$ high after $\overline{LAL}$ high	$2t_Q-20$		$2t_Q-10$		$2t_Q-10$		ns
58	$t_{h(RL\text{-}ALL)}$	Hold time of $\overline{LAL}$ low after $\overline{RAS}$ low	$6t_Q-20$		$6t_Q-10$		$6t_Q-10$		ns
59	$t_{h(CL\text{-}RL)}$	Hold time of $\overline{RAS}$ low after $\overline{CAS}$ low	$3t_Q-20$		$3t_Q-10$		$3t_Q-10$		ns
60	$t_{h(RH\text{-}WH)}$	Hold time of $\overline{W}$ high after after $\overline{RAS}$ high, shift register transfer follows read	$2t_Q-20$		$2t_Q-10$		$2t_Q-10$		ns
61	$t_{su(CAV\text{-}CL)}$	Setup time of column address valid to $\overline{CAS}\downarrow$	t_Q-20		t_Q-10		t_Q-10		ns
62	$t_{h(CL\text{-}ALL)}$	Hold time of $\overline{LAL}$ low after $\overline{CAS}$ low	$4t_Q-20$		$4t_Q-10$		$4t_Q-10$		ns
63	$t_{h(CH\text{-}ALL)}$	Hold time of $\overline{LAL}$ low after $\overline{CAS}$ high, write cycle	$0.5t_Q-15$		$0.5t_Q-10$		$0.5t_Q-10$		ns
64	$t_{h(ALL\text{-}CH)}$	Hold time of $\overline{CAS}$ high after $\overline{LAL}$ low	$0.5t_Q-15$		$0.5t_Q-10$		$0.5t_Q-10$		ns
65	$t_{h(CH\text{-}RH)}$	Hold time of $\overline{RAS}$ high after $\overline{CAS}$ high	$2.5t_Q-15$		$2.5t_Q-10$		$2.5t_Q-10$		ns
66	$t_{w(CL)}$	Pulse duration, $\overline{CAS}$ low	$3.5t_Q-25$		$3.5t_Q-10$		$3.5t_Q-10$		ns
67	$t_{h(RL\text{-}CH)}$	Hold time of $\overline{CAS}$ high after $\overline{RAS}$ low	$2t_Q-20$		$2t_Q-10$		$2t_Q-10$		ns
68	$t_{d(RL\text{-}CL)}$	Delay time from $\overline{RAS}$ low to $\overline{CAS}$ low	$2t_Q+20$		$2t_Q+10$		$2t_Q+10$		ns
69	$t_{h(CH\text{-}WH)}$	Hold time of $\overline{W}$ high after $\overline{CAS}$ high, shift register transfer follows read	$1.5t_Q-15$		$1.5t_Q-10$		$1.5t_Q-10$		ns
70	$t_{h(RL\text{-}CL)}$	Hold time of $\overline{CAS}$ low after $\overline{RAS}$ low	$5.5t_Q-25$		$5.5t_Q-10$		$5.5t_Q-10$		ns
71	$t_{w(CH)}$	Pulse duration, $\overline{CAS}$ high	$4.5t_Q-15$		$4.5t_Q-10$		$4.5t_Q-10$		ns
72	$t_{h(WH\text{-}ALL)}$	Hold time of $\overline{LAL}$ low after $\overline{W}$ high, write cycle	$0.5t_Q-15$		$0.5t_Q-10$		$0.5t_Q-10$		ns
73	$t_{su(WH\text{-}CL)}$	Setup time of $\overline{W}$ high to $\overline{CAS}\downarrow$, end of write	$4.5t_Q-15$		$4.5t_Q-10$		$4.5t_Q-10$		ns

† This is a functional minimum and is not tested.

local bus timing: the $\overline{RAS}$, $\overline{CAS}$, and $\overline{LAL}$ outputs



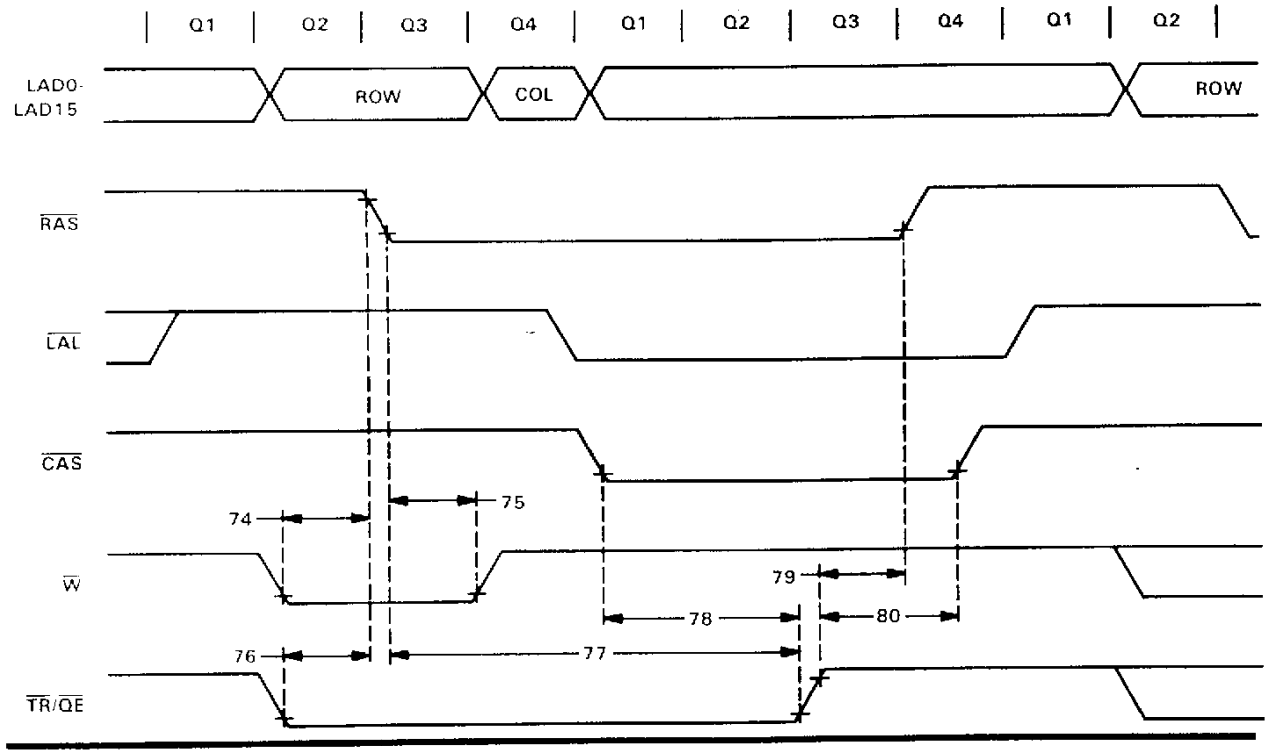
local bus timing parameters (continued)

Quarter clock time t_Q, which appears in the following table, is one quarter of a local output clock period, or $2t_{c(ICK)}$.

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
74	$t_{su(WL\text{-}RL)}$	Setup time of $\overline{W}$ low to $\overline{RAS}$1, serial register transfer cycle	$t_Q - 20$		t_Q 10		ns
75	$t_{h(RL\text{-}WL)}$	Hold time of $\overline{W}$ low after $\overline{RAS}$ low, serial register transfer cycle	$t_Q - 20$		$t_Q - 10$		ns
76	$t_{su(TRL\text{-}RL)}$	Setup time of $\overline{TR}/\overline{QE}$ low to $\overline{RAS}$1, serial register transfer cycle	$t_Q - 20$		$t_Q - 10$		ns
77	$t_{h(RL\text{-}TRL)}$	Hold time of $\overline{TR}/\overline{QE}$ low after $\overline{RAS}$ low, serial register transfer cycle	$4t_Q - 20$		$4t_Q - 10$		ns
78	$t_{h(CL\text{-}TRL)}$	Hold time of $\overline{TR}/\overline{QE}$ low after $\overline{CAS}$ low, serial register transfer cycle	$2t_Q - 20$		$2t_Q - 10$		ns
79	$t_{su(TRH\text{-}RH)}$	Setup time of $\overline{TR}/\overline{QE}$ high to $\overline{RAS}$1, serial register transfer cycle	$t_Q - 20$		$t_Q - 10$		ns
80	$t_{su(TRH\text{-}CH)}$	Setup time of $\overline{TR}/\overline{QE}$ high to $\overline{CAS}$1, serial register transfer cycle	$1.5t_Q - 25$		$1.5t_Q - 10$		ns

NOTE: Parameters 81 and 82 intentionally omitted.

local bus timing parameters: VRAM serial register transfer cycle



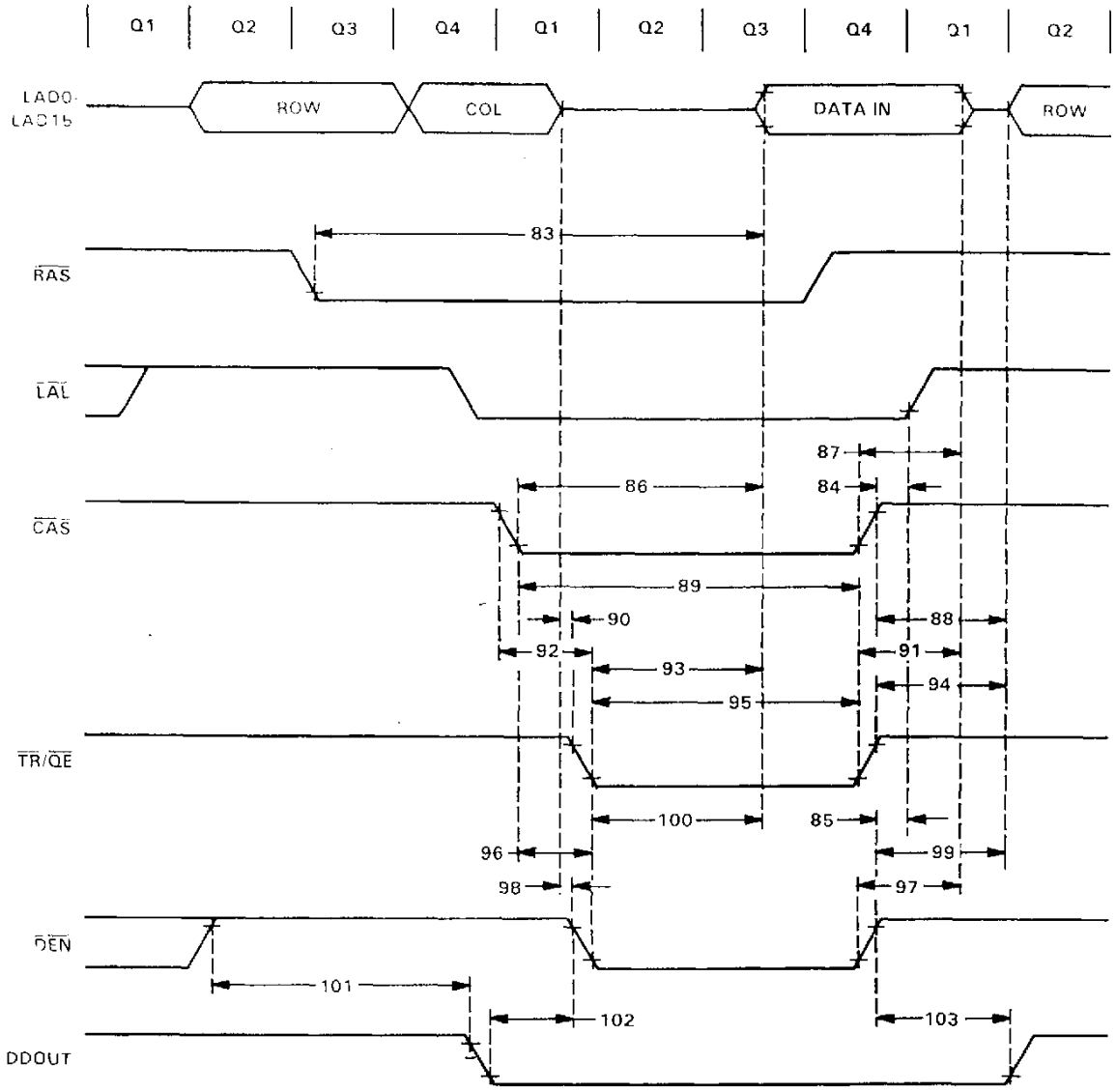
local bus timing parameters (continued)

Quarter clock time t_Q, which appears in the following table, is one quarter of a local output clock period, or $2t_{c(ICK)}$.

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
83	$t_{a(RL-DV)}$	Access time from $\overline{RAS}$ low to data in valid, read cycle		$5.5t_Q-40$†		$5.5t_Q-25$†	ns
84	$t_{su(CH-ALH)}$	Setup time of $\overline{CAS}$ high to $\overline{LAL}$↑	$0.5t_Q-15$		$0.5t_Q-10$		ns
85	$t_{su(ENH-ALH)}$	Setup time of $\overline{DEN}$ high to $\overline{LAL}$↑	$0.5t_Q-15$		$0.5t_Q-10$		ns
86	$t_{a(CL-DV)}$	Access time from $\overline{CAS}$ low to data in valid, read cycle		$3.5t_Q-40$†		$3.5t_Q-25$†	ns
87	$t_{h(CH-DV)}$	Hold time of data in valid after $\overline{CAS}$↑, read cycle	0		0		ns
88	$t_{h(CH-RAZ)}$	Hold time of row address high impedance after $\overline{CAS}$ high, end of read cycle	$1.5t_Q-10$‡		$1.5t_Q-10$‡		ns
89	$t_{h(CL-QEL)}$	Hold time of $\overline{TR}/\overline{QE}$ low after $\overline{CAS}$ low, read cycle	$3.5t_Q-25$		$3.5t_Q-10$		ns
90	$t_{su(CAZ-QEL)}$	Setup time of column address high impedance to $\overline{TR}/\overline{QE}$↓, read cycle	t_Q-10‡		t_Q-10‡		ns
91	$t_{h(QEH-DV)}$	Hold time of data in valid after $\overline{TR}/\overline{QE}$↑, read cycle	0		0		ns
92	$t_{d(CL-QEL)}$	Delay time from $\overline{CAS}$↓ to $\overline{TR}/\overline{QE}$ low, read cycle		t_Q+20		t_Q+10	ns
93	$t_{a(QEL-DV)}$	Access time from $\overline{TR}/\overline{QE}$ low to data in valid, read cycle		$2.5t_Q-40$†		$2.5t_Q-25$†	ns
94	$t_{h(QEH-RAZ)}$	Hold time of row address high impedance after $\overline{TR}/\overline{QE}$ high, end of read cycle	$1.5t_Q-10$‡		$1.5t_Q-10$‡		ns
95	$t_{w(QEL)}$	Pulse duration, $\overline{TR}/\overline{QE}$ low, read cycle	$2.5t_Q-25$		$2.5t_Q-10$		ns
96	$t_{d(CL-ENL)}$	Delay time from $\overline{CAS}$ low to $\overline{DEN}$ low, read cycle		t_Q+20		t_Q+10	ns
97	$t_{h(ENH-DV)}$	Hold time of data in valid after $\overline{DEN}$↑, read cycle	0		0		ns
98	$t_{su(CAZ-ENL)}$	Setup time of column address high impedance to $\overline{DEN}$↓, read cycle	t_Q-10‡		t_Q-10‡		ns
99	$t_{h(ENH-RAZ)}$	Hold time of next row address high impedance after $\overline{DEN}$ high, end of read cycle	$1.5t_Q-10$‡		$1.5t_Q-10$‡		ns
100	$t_{a(ENL-DV)}$	Access time from $\overline{DEN}$ low to data in valid, read cycle		$2.5t_Q-40$†		$2.5t_Q-25$†	ns
101	$t_{h(ENH-DDH)}$	Hold time of DDOUT high after $\overline{DEN}$ high, read follows write cycle	$3t_Q-20$		$3t_Q-10$		ns
102	$t_{su(DDL-ENL)}$	Setup time of DDOUT low to $\overline{DEN}$↓, read cycle	t_Q-20		t_Q-10		ns

†4_{tQ} is added to these values for each wait state inserted.
‡These values are derived from characterization and are not tested.

local bus timing: read cycle

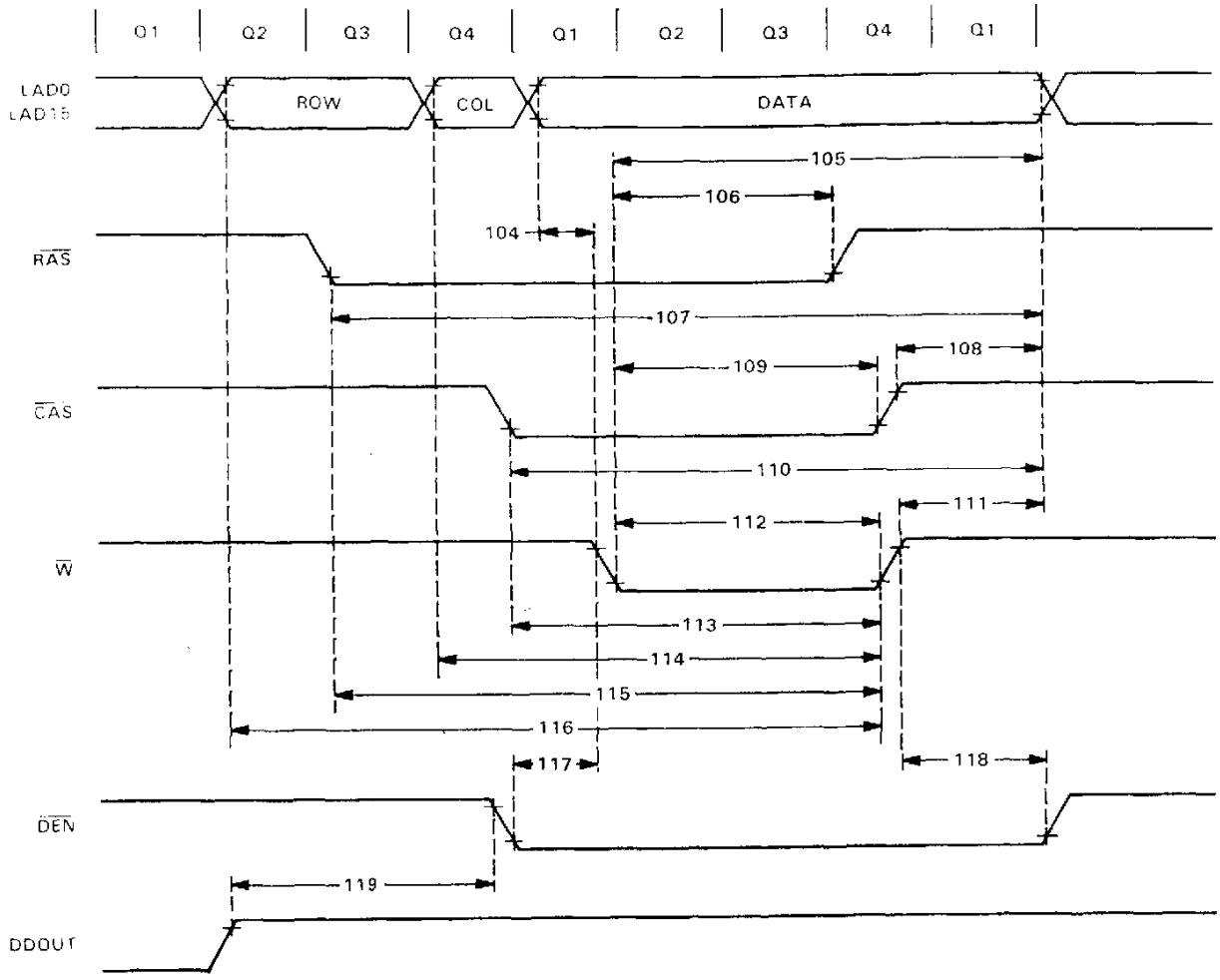


local bus timing parameters (continued)

Quarter clock time t_Q, which appears in the following table, is one quarter of a local output clock period, or $2t_{c(ICK)}$.

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
103	$t_{h(ENH-DDL)}$	Hold time of DDOUT low after $\overline{DEN}$ high, read cycle	$1.5t_Q-15$		$1.5t_Q-10$		ns
104	$t_{su(DV-WL)}$	Setup time of data out valid to $\overline{W}$↓, write cycle	t_Q-20		t_Q-15		ns
105	$t_{h(WL-DV)}$	Hold time of data out valid after $\overline{W}$ low, write cycle	$4t_Q-20$		$4t_Q-10$		ns
106	$t_{su(WL-RH)}$	Setup time of $\overline{W}$ low to $\overline{RAS}$↑, write cycle	$2t_Q-20$		$2t_Q-10$		ns
107	$t_{h(RL-DV)}$	Hold time of data out valid after $\overline{RAS}$ low, write cycle	$7t_Q-20$		$7t_Q-10$		ns
108	$t_{h(CH-DV)}$	Hold time of data out valid after $\overline{CAS}$ high, write cycle	$1.5t_Q-15$		$1.5t_Q-10$		ns
109	$t_{su(WL-CH)}$	Setup time of $\overline{W}$ low to $\overline{CAS}$↑, write cycle	$2.5t_Q-25$		$2.5t_Q-10$		ns
110	$t_{h(CL-DV)}$	Hold time of data out valid after $\overline{CAS}$ low, write cycle	$5t_Q-20$		$5t_Q-10$		ns
111	$t_{h(WH-DV)}$	Hold time of data out valid after $\overline{W}$ high, write cycle	$1.5t_Q-15$		$1.5t_Q-10$		ns
112	$t_{w(WL)}$	Pulse duration, $\overline{W}$ low	$2.5t_Q-25$		$2.5t_Q-10$		ns
113	$t_{h(CL-WL)}$	Hold time of $\overline{W}$ low after $\overline{CAS}$ low, write cycle	$3.5t_Q-25$		$3.5t_Q-10$		ns
114	$t_{su(CAV-WH)}$	Setup time of column address valid to $\overline{W}$↑, write cycle	$4.5t_Q-30$		$4.5t_Q-15$		ns
115	$t_{h(RL-WL)}$	Hold time of $\overline{W}$ low after $\overline{RAS}$ low, write cycle	$5.5t_Q-25$		$5.5t_Q-10$		ns
116	$t_{su(RAV-WH)}$	Setup time of row address valid to $\overline{W}$↑, write cycle	$6.5t_Q-35$		$6.5t_Q-15$		ns
117	$t_{su(ENL-WL)}$	Setup time of $\overline{DEN}$ low to $\overline{W}$↓, write cycle	t_Q-20		t_Q-10		ns
118	$t_{h(WH-ENL)}$	Hold time of $\overline{DEN}$ low after $\overline{W}$ high, write cycle	$1.5t_Q-15$		$1.5t_Q-10$		ns
119	$t_{su(DDH-ENL)}$	Setup time of DDOUT high to $\overline{DEN}$↓, write follows read	$3t_Q-20$		$3t_Q-10$		ns

local bus timing: write cycle



local bus timing parameters (continued)

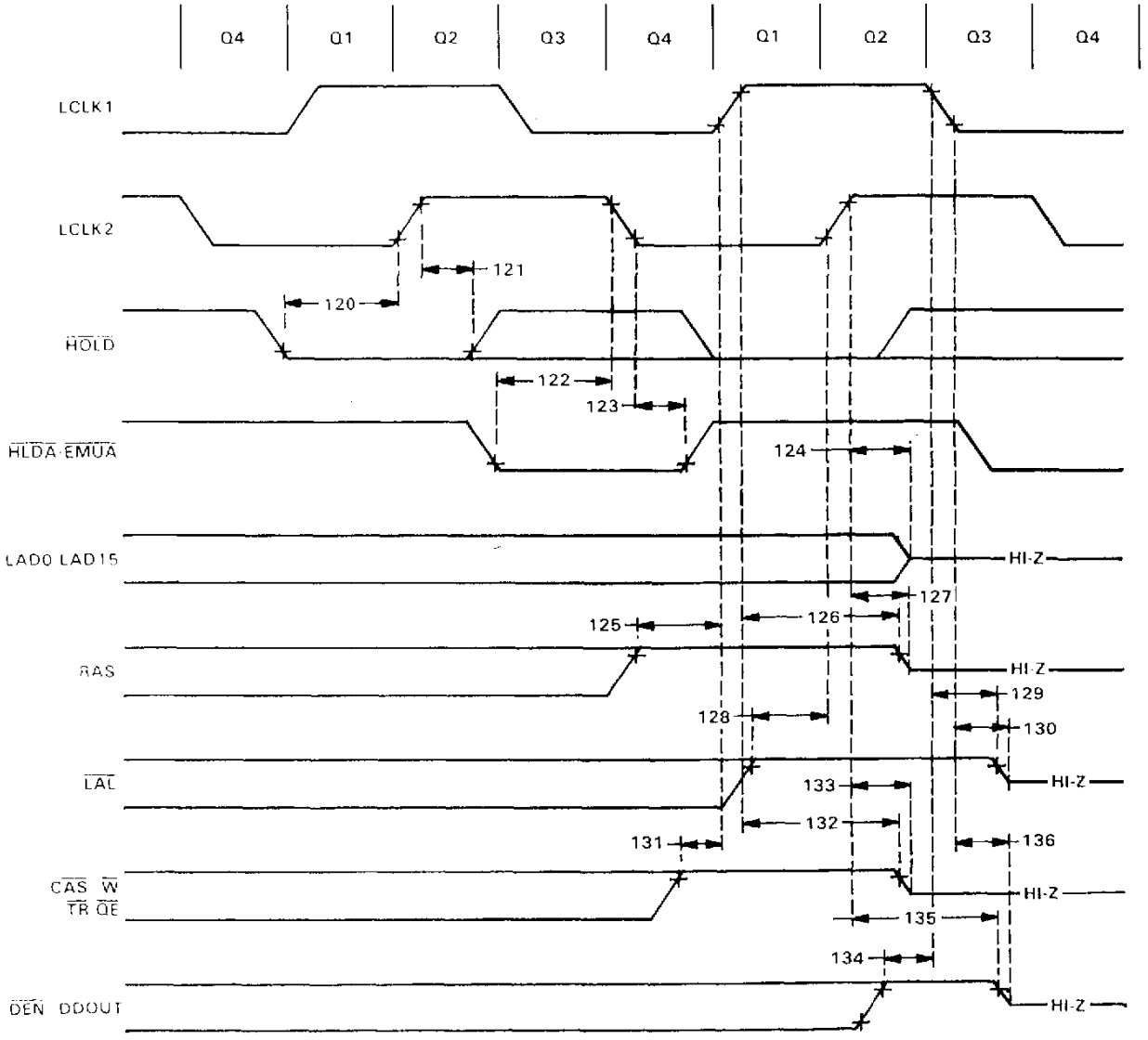
Quarter clock time t_Q, which appears in the following table, is one quarter of a local output clock period, or $2t_{c(ICK)}$.

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
120	$t_{su(HRV-CK2H)}$	Setup time of $\overline{HOLD}$ valid to LCLK2↑	50†		40†		ns
121	$t_{h(CK2H-HRV)}$	Hold time of $\overline{HOLD}$ valid after LCLK2 high	0†		0†		ns
122	$t_{su(HKV-CK2L)}$	Setup time of $\overline{HLDA}/\overline{EMUA}$ output valid before LCLK2↓	t_Q-20		t_Q-10		ns
123	$t_{h(CK2L-HKL)}$	Hold time of $\overline{HLDA}/\overline{EMUA}$ low, after LCLK2 low	t_Q-15		t_Q-15		ns
124	$t_{d(CK2H-DZ)}$	Delay from LCLK2 high to LAD pins high impedance, bus release		30‡		30‡	ns
125	$t_{su(RH-CK1H)}$	Setup time of $\overline{RAS}$ high to LCLK1↑	t_Q-20		t_Q-10		ns
126	$t_{h(CK1H-RH)}$	Hold time of $\overline{RAS}$ driven high after LCLK1 high, bus release	t_Q-10‡		t_Q-10‡		ns
127	$t_{d(CK2H-RZ)}$	Delay from LCLK2 high to $\overline{RAS}$ high impedance, bus release		30‡		30‡	ns
128	$t_{su(ALH-CK2H)}$	Setup time of $\overline{LAL}$ high to LCLK2↑	t_Q-20		t_Q-10		ns
129	$t_{h(CK1L-ALH)}$	Hold time of $\overline{LAL}$ driven high after LCLK1↓, bus release	−5‡		−5‡		ns
130	$t_{d(CK1L-ALZ)}$	Delay from LCLK1 low to $\overline{LAL}$ high impedance, bus release		30‡		30‡	ns
131	$t_{su(CH-CK1H)}$	Setup time of $\overline{CAS}$, $\overline{W}$, and $\overline{TR}/\overline{QE}$ high to LCLK1↑	$0.5t_Q-15$		$0.5t_Q-10$		ns
132	$t_{h(CK1H-CH)}$	Hold time of $\overline{CAS}$, $\overline{W}$, and $\overline{TR}/\overline{QE}$ high after LCLK1 high, bus release	t_Q-10‡		t_Q-10‡		ns
133	$t_{d(CK2H-CZ)}$	Delay from LCLK2 high to $\overline{CAS}$, $\overline{W}$, and $\overline{TR}/\overline{QE}$ high impedance, bus release		30‡		30‡	ns
134	$t_{su(ENH-CK2H)}$	Setup time of $\overline{DEN}$ or DDOUT high to LCLK1↑	t_Q-20		t_Q-10		ns
135	$t_{h(CK2H-ENH)}$	Hold time of $\overline{DEN}$ and DDOUT high after LCLK1↓, bus release	t_Q-10‡		t_Q-10‡		ns
136	$t_{d(CK1L-ENZ)}$	Delay from LCLK1 low to $\overline{DEN}$ and DDOUT high impedance, bus release		30‡		30‡	ns
137	$t_{h(CK2H-DZ)}$	Hold time of LAD bus high impedance after LCLK2↑	−5‡		−5‡		ns
138	$t_{h(CK2H-RZ)}$	Hold time of $\overline{RAS}$, $\overline{CAS}$, $\overline{W}$, $\overline{LAL}$, and $\overline{TR}/\overline{QE}$ high impedance after LCLK1↑	−5‡		−5‡		ns
139	$t_{d(CK1H-RH)}$	Delay from LCLK1 high to RAS, $\overline{CAS}$, $\overline{W}$, $\overline{LAL}$, and $\overline{TR}/\overline{QE}$ driven high, resume bus control		30		30	ns
140	$t_{h(CK2H-RH)}$	Hold time of $\overline{RAS}$ high after LCLK2 high, resumes bus control	t_Q-15		t_Q-10		ns
141	$t_{h(CK2H-CH)}$	Hold time of $\overline{CAS}$, $\overline{W}$, and $\overline{TR}/\overline{QE}$ high after LCLK2 high, resume bus control	5‡		−5‡		ns

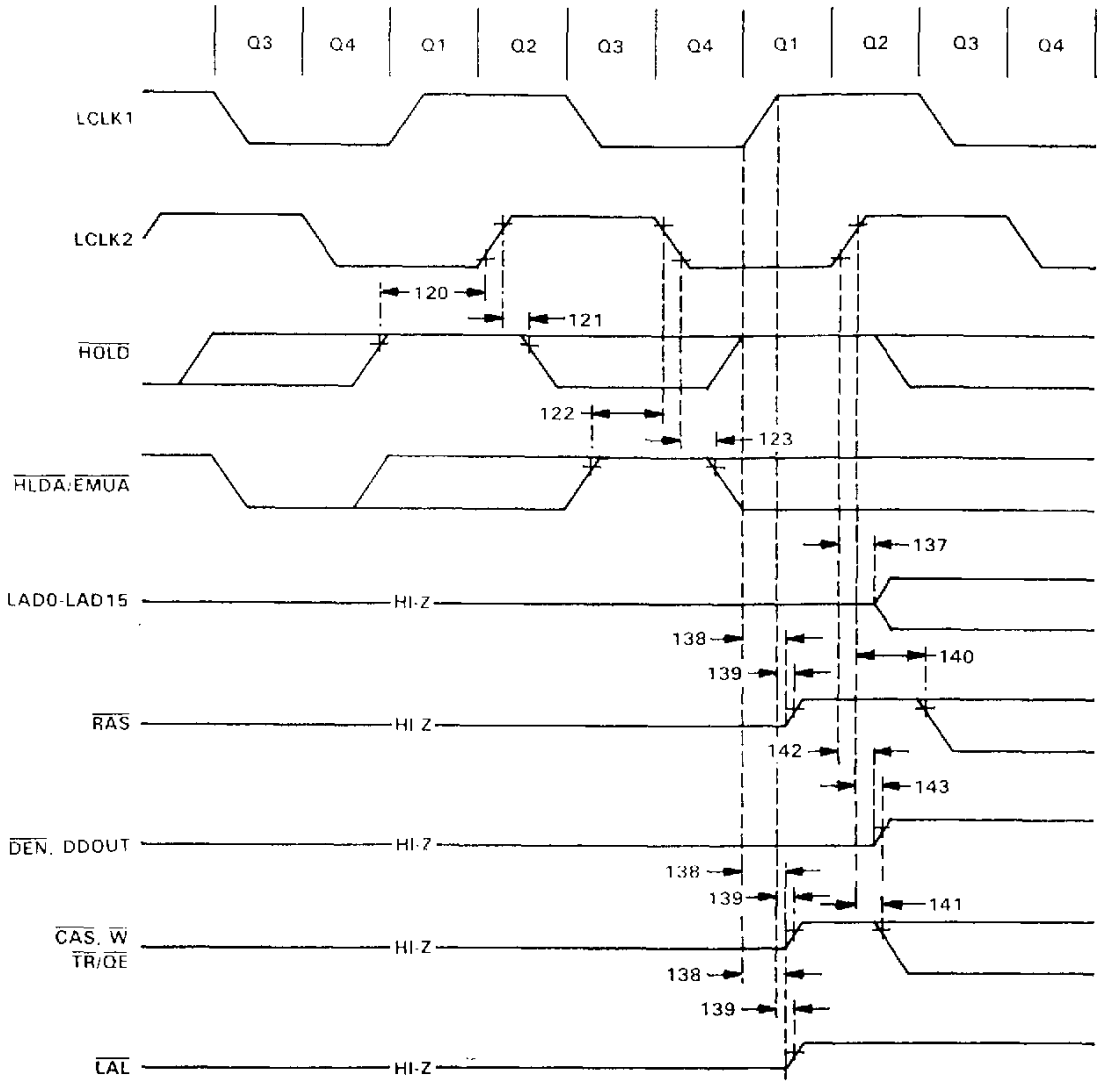
† $\overline{HOLD}$ is a synchronous input sampled during the low-to-high transition of LCLK2. The specified setup and hold times must be met for the device to operation properly.

‡ These values are derived from characterization and are not tested.

GSP releases control of local bus



GSP resumes control of local bus



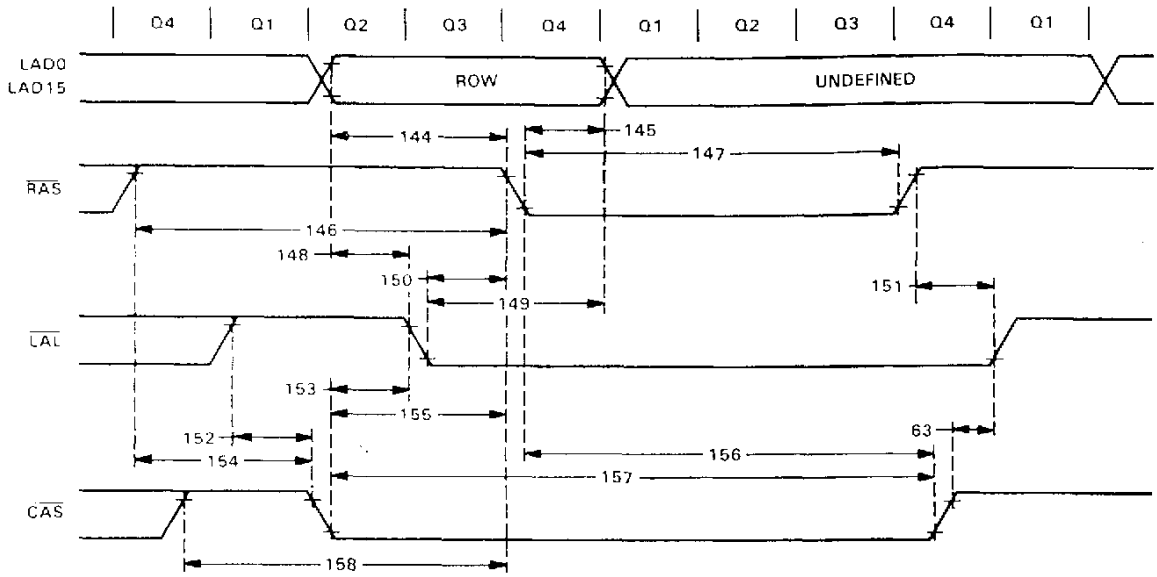
local bus timing parameters (continued)

Quarter clock time t_Q, which appears in the following table, is one quarter of a local output clock cycle, or $2t_{c(ICK)}$.

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
142	$t_{h(CK2H-ENZ)}$	Hold time of $\overline{DEN}$, DDOUT high impedance after LCLK2 high, resume bus control	−5†		−5†		ns
143	$t_{d(CK2H-ENH)}$	Delay from LCLK2 high to $\overline{DEN}$, and DDOUT driven high, resume bus control		30		30	ns
144	$t_{su(RAV-RL)}$	Setup time of row address valid to $\overline{RAS}$↓, CAS-before-RAS refresh	$2t_Q-25$		$2t_Q-15$		ns
145	$t_{h(RL-RAV)}$	Hold time of row address valid after $\overline{RAS}$ low, $\overline{CAS}$-before-$\overline{RAS}$ refresh	t_Q-20		t_Q-10		ns
146	$t_{w(RH)}$	Pulse duration, $\overline{RAS}$ high, start of $\overline{CAS}$-before-$\overline{RAS}$ refresh	$4t_Q-20$		$4t_Q-10$		ns
147	$t_{w(RL)}$	Pulse duration, $\overline{RAS}$ low, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$4t_Q-20$		$4t_Q-10$		ns
148	$t_{su(RAV-ALL)}$	Setup time of row address valid to $\overline{LAL}$↓, $\overline{CAS}$-before-$\overline{RAS}$ refresh	t_Q-20		t_Q-15		ns
149	$t_{h(ALL-RAV)}$	Hold time of row address valid after $\overline{LAL}$ low, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$2t_Q-20$		$2t_Q-10$		ns
150	$t_{h(ALL-RH)}$	Hold time of $\overline{RAS}$ high after $\overline{LAL}$ low, $\overline{CAS}$-before-$\overline{RAS}$ refresh	t_Q-20		t_Q-10		ns
151	$t_{su(RH-ALH)}$	Setup time of $\overline{RAS}$ high to $\overline{LAL}$↑, $\overline{CAS}$-before-$\overline{RAS}$ refresh	t_Q-20		t_Q-10		ns
152	$t_{su(ALH-CL)}$	Setup time of $\overline{LAL}$ high to $\overline{CAS}$↓, $\overline{CAS}$-before-$\overline{RAS}$ refresh	t_Q-20		t_Q-10		ns
153	$t_{su(CL-ALL)}$	Setup time of $\overline{CAS}$ low to $\overline{LAL}$↓, $\overline{CAS}$-before-$\overline{RAS}$ refresh	t_Q-20		t_Q-10		ns
154	$t_{su(RH-CL)}$	Setup time of $\overline{RAS}$ high to $\overline{CAS}$↓, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$2t_Q-20$		$2t_Q-10$		ns
155	$t_{su(CL-RL)}$	Setup time of $\overline{CAS}$ low to $\overline{RAS}$↓, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$2t_Q-20$		$2t_Q-10$		ns
156	$t_{h(RL-CL)}$	Hold time of $\overline{CAS}$ low after $\overline{RAS}$ low, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$4.5t_Q-25$		$4.5t_Q-10$		ns
157	$t_{w(CL)}$	Pulse duration, $\overline{CAS}$ low, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$6.5t_Q-25$		$6.5t_Q-10$		ns
158	$t_{su(CH-RL)}$	Setup time of $\overline{CAS}$ high to $\overline{RAS}$↓, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$3.5t_Q-15$		$3.5t_Q-10$		ns

†These values are derived from characterization and are not tested.

$\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ DRAM refresh cycle timing



local bus timing parameters (continued)

Quarter clock time t_Q, which appears in the following table, is one quarter of a local output clock cycle, or $2t_{c(ICK)}$.

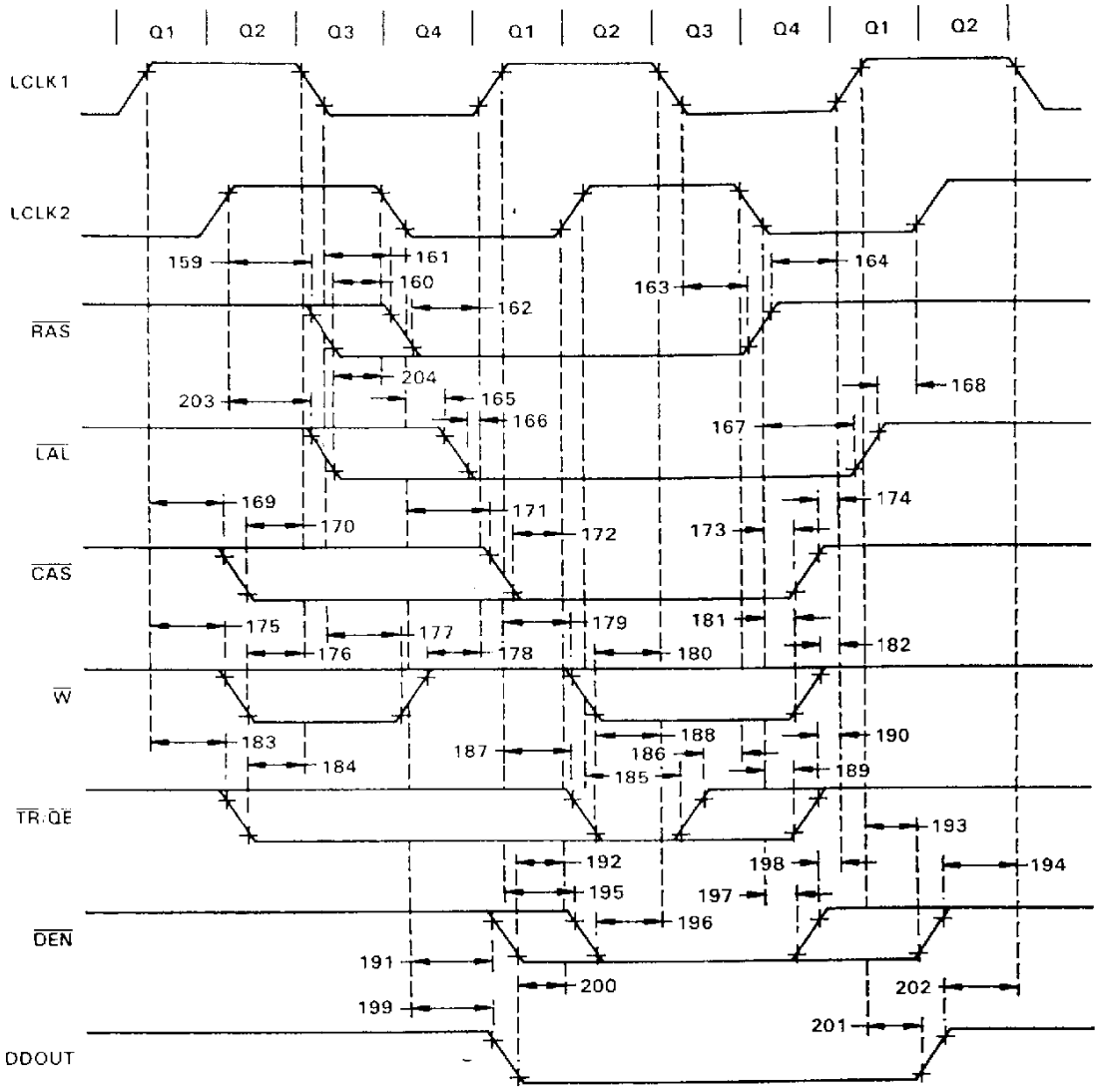
NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
159	$t_{h(CK2H-RH)}$	Hold time of $\overline{RAS}$ high after LCLK2 high, all cycles except internal and $\overline{CAS}$-before-$\overline{RAS}$ refresh	$t_Q - 15$		$t_Q - 10$		ns
160	$t_{su(RL-CK2L)}$	Setup time of $\overline{RAS}$ low to LCLK2↓, all cycles except internal and $\overline{CAS}$ before-$\overline{RAS}$ refresh	$t_Q - 20$		$t_Q - 10$		ns
161	$t_{h(CK1L-RH)}$	Hold time of $\overline{RAS}$ high after LCLK1 low, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$t_Q - 15$		$t_Q - 10$		ns
162	$t_{su(RL-CK1H)}$	Setup time of $\overline{RAS}$ low to LCLK1↑, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$t_Q - 20$		$t_Q - 10$		ns
163	$t_{h(CK1L-RL)}$	Hold time of $\overline{RAS}$ low after LCLK1 low, all cycles except internal	$t_Q - 15$		$t_Q - 10$		ns
164	$t_{su(RH-CK1H)}$	Setup time of $\overline{RAS}$ high to LCLK1↑, all cycles except internal	$t_Q - 20$		$t_Q - 10$		ns
165	$t_{h(CK2L-ALH)}$	Hold time of $\overline{LAL}$ high after LCLK2 low, all cycles except internal	$0.5t_Q - 15$		$0.5t_Q - 10$		ns
166	$t_{su(ALL-CK1H)}$	Setup time of $\overline{LAL}$ low to LCLK1↑, all cycles except internal	$0.5t_Q - 15$		$0.5t_Q - 10$		ns
167	$t_{h(CK2L-ALL)}$	Hold time of $\overline{LAL}$ low after LCLK2 low, all cycles except internal	$t_Q - 15$		$t_Q - 10$		ns
168	$t_{su(ALH-CK2H)}$	Setup time of $\overline{LAL}$ high to LCLK2↑, all cycles except internal	$t_Q - 20$		$t_Q - 10$		ns
169	$t_{h(CK1H-CH)}$	Hold time of $\overline{CAS}$ high after LCLK1 high, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$t_Q - 15$		$t_Q - 10$		ns
170	$t_{su(CL-CK1L)}$	Setup time of $\overline{CAS}$ low to LCLK1↓, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$t_Q - 20$		$t_Q - 10$		ns
171	$t_{h(CK2L-CH)}$	Hold time of $\overline{CAS}$ high after LCLK2 low, cycles except internal, DRAM refresh and $\overline{CAS}$-before-$\overline{RAS}$ refresh	$t_Q - 15$		$t_Q - 10$		ns
172	$t_{su(CL-CK2H)}$	Setup time of $\overline{CAS}$ low to LCLK2↑, all cycles except internal, DRAM refresh, and $\overline{CAS}$-before-$\overline{RAS}$ refresh	$t_Q - 20$		$t_Q - 10$		ns
173	$t_{h(CK2L-CL)}$	Hold time of $\overline{CAS}$ low after LCLK2 low, all cycles except internal and DRAM refresh	$0.5t_Q - 15$		$0.5t_Q - 10$		ns
174	$t_{su(CH-CK1H)}$	Setup time of $\overline{CAS}$ high to LCLK1↑, all cycles except internal and DRAM refresh	$0.5t_Q - 15$		$0.5t_Q - 10$		ns
175	$t_{h(CK1H-WH)TR}$	Hold time of $\overline{W}$ high after LCLK1 high, shift register transfer	$t_Q - 15$		$t_Q - 10$		ns
176	$t_{su(WL-CK1L)TR}$	Setup time of $\overline{W}$ low to LCLK1↓, shift register transfer	$t_Q - 20$		$t_Q - 10$		ns

local bus timing parameters (concluded)

Quarter clock time t_Q, which appears in the following table, is one quarter of a local output clock cycle, or $2t_{c(ICK)}$.

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
177	$t_{h(CK1L-WL)}$	Hold time of $\overline{W}$ low after LCLK1 low, shift register transfer	$t_Q - 15$		$t_Q - 10$		ns
178	$t_{su(WH-CK1H)}$	Setup time of $\overline{W}$ high to LCLK1↑, shift register transfer	$t_Q - 20$		$t_Q - 10$		ns
179	$t_{h(CK1H-WH)}$	Hold time of $\overline{W}$ high after LCLK1 high, write	$t_Q - 15$		$t_Q - 10$		ns
180	$t_{su(WL-CK1L)}$	Setup time of $\overline{W}$ low to LCLK1↓, write	$t_Q - 20$		$t_Q - 10$		ns
181	$t_{h(CK2L-WL)}$	Hold time of $\overline{W}$ low after LCLK2 low, write	$0.5t_Q - 15$		$0.5t_Q - 10$		ns
182	$t_{su(WH-CK1H)}$	Setup time of $\overline{W}$ high to LCLK1↑, write	$0.5t_Q - 15$		$0.5t_Q - 10$		ns
183	$t_{h(CK1L-TRH)}$	Hold time of $\overline{TR}/\overline{QE}$ high after LCLK1 high, shift register transfer	$t_Q - 15$		$t_Q - 10$		ns
184	$t_{su(TRL-CK1H)}$	Setup time of $\overline{TR}/\overline{QE}$ low to LCLK1↓, shift register transfer	$t_Q - 20$		$t_Q - 10$		ns
185	$t_{h(CK2H-TRL)}$	Hold time of $\overline{TR}/\overline{QE}$ low after LCLK2 high, shift register transfer	$t_Q - 15$		$t_Q - 10$		ns
186	$t_{su(TRH-CK2L)}$	Setup time of $\overline{TR}/\overline{QE}$ high to LCLK2↓, shift register transfer	$t_Q - 20$		$t_Q - 10$		ns
187	$t_{h(CK1H-QEH)}$	Hold time of $\overline{TR}/\overline{QE}$ high after LCLK1 high, read	$t_Q - 15$		$t_Q - 10$		ns
188	$t_{su(QEL-CK1L)}$	Setup time of $\overline{TR}/\overline{QE}$ low to LCLK1↓, read	$t_Q - 20$		$t_Q - 10$		ns
189	$t_{h(CK2L-QEL)}$	Hold time of $\overline{TR}/\overline{QE}$ low after LCLK2 low, read	$0.5t_Q - 15$		$0.5t_Q - 10$		ns
190	$t_{su(QEH-CK1H)}$	Setup time of $\overline{TR}/\overline{QE}$ high to LCLK1↑, read	$0.5t_Q - 15$		$0.5t_Q - 10$		ns
191	$t_{h(CK2L-ENH)}$	Hold time of $\overline{DEN}$ high after LCLK2 low, write	$t_Q - 15$		$t_Q - 10$		ns
192	$t_{su(ENL-CK2H)}$	Setup time of $\overline{DEN}$ low to LCLK2↑, read	$t_Q - 20$		$t_Q - 10$		ns
193	$t_{h(CK1H-ENL)}$	Hold time of $\overline{DEN}$ low after LCLK1 high, write	$t_Q - 15$		$t_Q - 10$		ns
194	$t_{su(ENH-CK1L)}$	Setup time of $\overline{DEN}$ high to LCLK1↓, write	$t_Q - 20$		$t_Q - 10$		ns
195	$t_{h(CK1H-ENH)}$	Hold time of $\overline{DEN}$ high after LCLK1 high, read	$t_Q - 15$		$t_Q - 10$		ns
196	$t_{su(ENL-CK1L)}$	Setup time of $\overline{DEN}$ low to LCLK1↓, read	$t_Q - 20$		$t_Q - 10$		ns
197	$t_{h(CK2L-ENL)}$	Hold time of $\overline{DEN}$ low after LCLK2 low, read	$0.5t_Q - 15$		$0.5t_Q - 10$		ns
198	$t_{su(ENH-CK1H)}$	Setup time of $\overline{DEN}$ high to LCLK1↑, read	$0.5t_Q - 15$		$0.5t_Q - 10$		ns
199	$t_{h(CK2L-DDH)}$	Hold time of DDOUT high after LCLK2 low, read	$t_Q - 15$		$t_Q - 10$		ns
200	$t_{su(DDL-CK2H)}$	Setup time of DDOUT low to LCLK2↑, read	$t_Q - 20$		$t_Q - 10$		ns
201	$t_{h(CK1H-DDL)}$	Hold time of DDOUT low after LCLK1 high, read	$t_Q - 15$		$t_Q - 10$		ns
202	$t_{su(DDH-CK1L)}$	Setup time of DDOUT high to LCLK1↓, read	$t_Q - 20$		$t_Q - 10$		ns
203	$t_{h(CK2H-ALH)}$	Hold time of $\overline{LAL}$ high after LCLK2 high, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$t_Q - 15$		$t_Q - 10$		ns
204	$t_{su(ALL-CK2L)}$	Setup time of $\overline{LAL}$ low to LCLK2↓, $\overline{CAS}$-before-$\overline{RAS}$ refresh	$t_Q - 20$		$t_Q - 10$		ns

local bus timing: relationship of control signals to clocks



video interface timing parameters

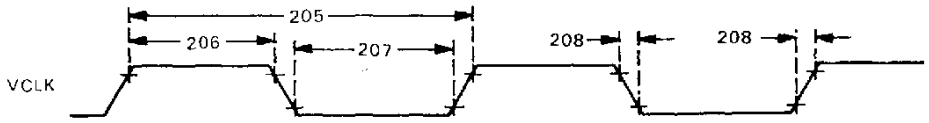
The timing parameters for TMS34010 video interface signals are shown in the next three tables and diagrams. The video interface includes the following TMS34010 pins: VCLK (video input clock), $\overline{\text{BLANK}}$ (blanking), $\overline{\text{HSYNC}}$ (horizontal sync, bidirectional), and $\overline{\text{VSYNC}}$ (vertical sync, bidirectional). HSYNC and $\overline{\text{VSYNC}}$ are inputs if external sync mode is enabled; otherwise they are outputs.

video input clock timing parameters

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
205	$t_{c(VCK)}$	Period of video input clock VCLK	100		80		ns
206	$t_{w(VCKH)}$	Pulse duration of VCLK high	40		30		ns
207	$t_{w(VCKL)}$	Pulse duration of VCLK low	40		30		ns
208	$t_{t(VCK)}$	Transition time (rise and fall) of VCLK		5†		5†	ns

†This value is determined through computer simulation and is not tested.

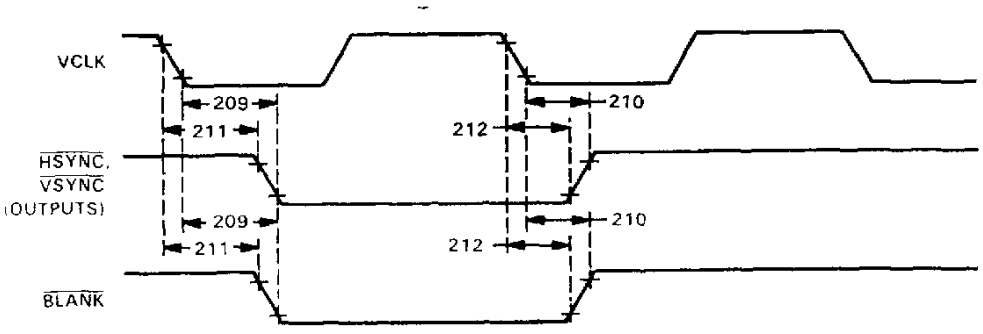
video input clock timing



video interface timing parameters: outputs

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
209	$t_{d(VCKL-HSL)}$	Delay from VCLK low to $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, or $\overline{\text{BLANK}}$ low		30		30	ns
210	$t_{d(VCKL-HSH)}$	Delay from VCLK low to $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, or $\overline{\text{BLANK}}$ high		30		30	ns
211	$t_{h(VCKL-HSH)}$	Hold time of $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, or $\overline{\text{BLANK}}$ high after VCLK↓	0		0		ns
212	$t_{h(VCKL-HSL)}$	Hold time of $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, or $\overline{\text{BLANK}}$ low after VCLK↓	0		0		ns

video output timing

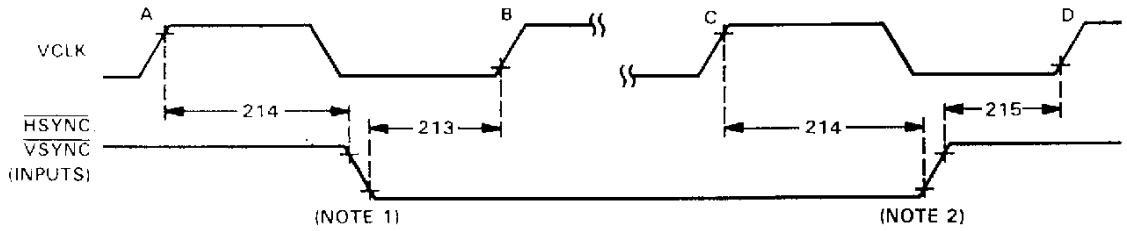


video interface timing: external sync inputs

NO.	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
213	$t_{su(HSV-VCKH)}$	Setup time of $\overline{HSYNC}$, $\overline{VSYNC}$ valid to VCLK↑	20†		20†		ns
214	$t_{h(VCKH-HSV)}$	Hold time of $\overline{HSYNC}$, $\overline{VSYNC}$ valid after VCLK high	20†		20†		ns
215	$t_{su(HSH-VCKH)}$	Setup time of $\overline{HSYNC}$, $\overline{VSYNC}$ high to VCLK↑	20‡		20‡		ns

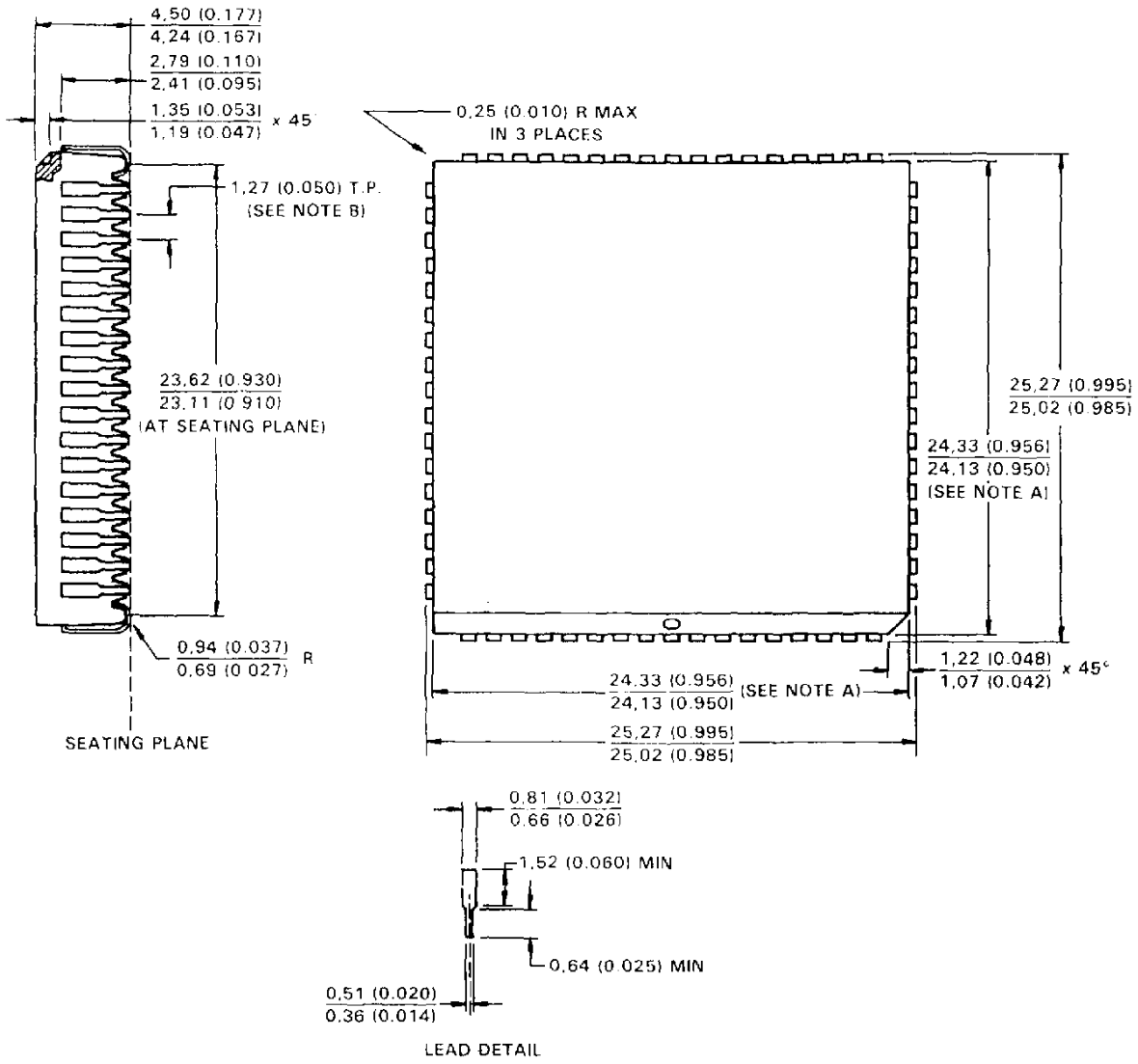
†Specified setup and hold times on asynchronous inputs are required only to guarantee recognition at indicated clock edge.
‡This value is determined through computer simulation.

external sync input timing



NOTES 1. If the falling edge of the sync signal occurs more than $t_{h(SV-VCH)}$ past VCLK edge A, and at least $t_{su(SV-VCH)}$ before edge B, the transition will be detected at edge B instead of edge A.
2. If the rising edge of the sync signal occurs more than $t_{h(SV-VCH)}$ past VCLK edge C, and at least $t_{su(SV-VCH)}$ before edge D, the transition will be detected at edge D instead of edge C.

MECHANICAL DATA



NOTES: A. Centerline of center pin each side is within 0.10 (0.004) of package centerline as determined by this dimension.
B. Location of each pin is within 0.127 (0.005) of true position with respect to center pin on each side.

ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES.

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.