TMC2302A

Image Manipulation Sequencer

40 MHz

Features

- Asynchronous loading of control parameters
- Rapid (25ns per pixel) rotation, warping, panning, and scaling of images
- Three-dimensional image addressing capability
- General third-order polynomial transformations in two dimensions on-chip
- Three-dimensional transformation of up to order 1.5 also supported
- Flexible, user-configurable pixel datapath timing structure
- Static convolutional filtering of up to 16 x 16 Pixel (one-pass), 256 x 256 pixel (two-pass) or 256 x 256 x 256 pixel (three-pass) windows
- User-selectable source image subpixel resolution of 2^{-8} to 2^{-16}
- Pin-compatible upgrade to TMC2302
- 24-bit (optional 36-bit) positioning precision within the source image space, 48-bit internal precision
- Low power CMOS process
- Available in a 120-pin Plastic Pin Grid Array and 120-lead Metric Quad Flat Pack

Applications

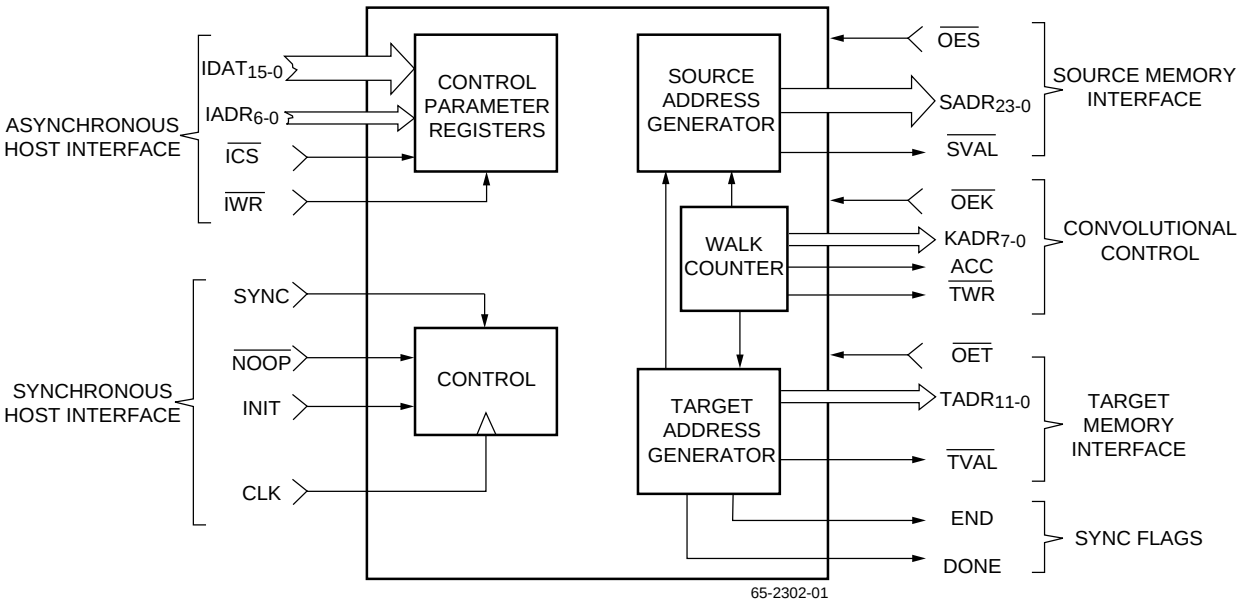
- High-performance video special-effects generators
- Guidance systems
- Image recognition
- Robotics
- High-precision image registration

Description

The TMC2302A, a pin-compatible replacement for the TMC2302, is a high-speed self-sequencing address generator which supports image manipulations such as rotation, rescaling, warping, filtering, and resampling. It remaps the pixel locations of a target (display) space back into those of a source image space. The degree and type of image manipulation is determined by the remapping selected.

To remap from the target to the source space, this integrated circuit computes a series of polynomials of the target space coordinates, based on user-assigned coefficients. Two TMC2302A chips can generate third-order warps of a two-dimensional image, whereas three can second-order warp a three-dimensional image.

Simplified Block Diagram



Preliminary Information

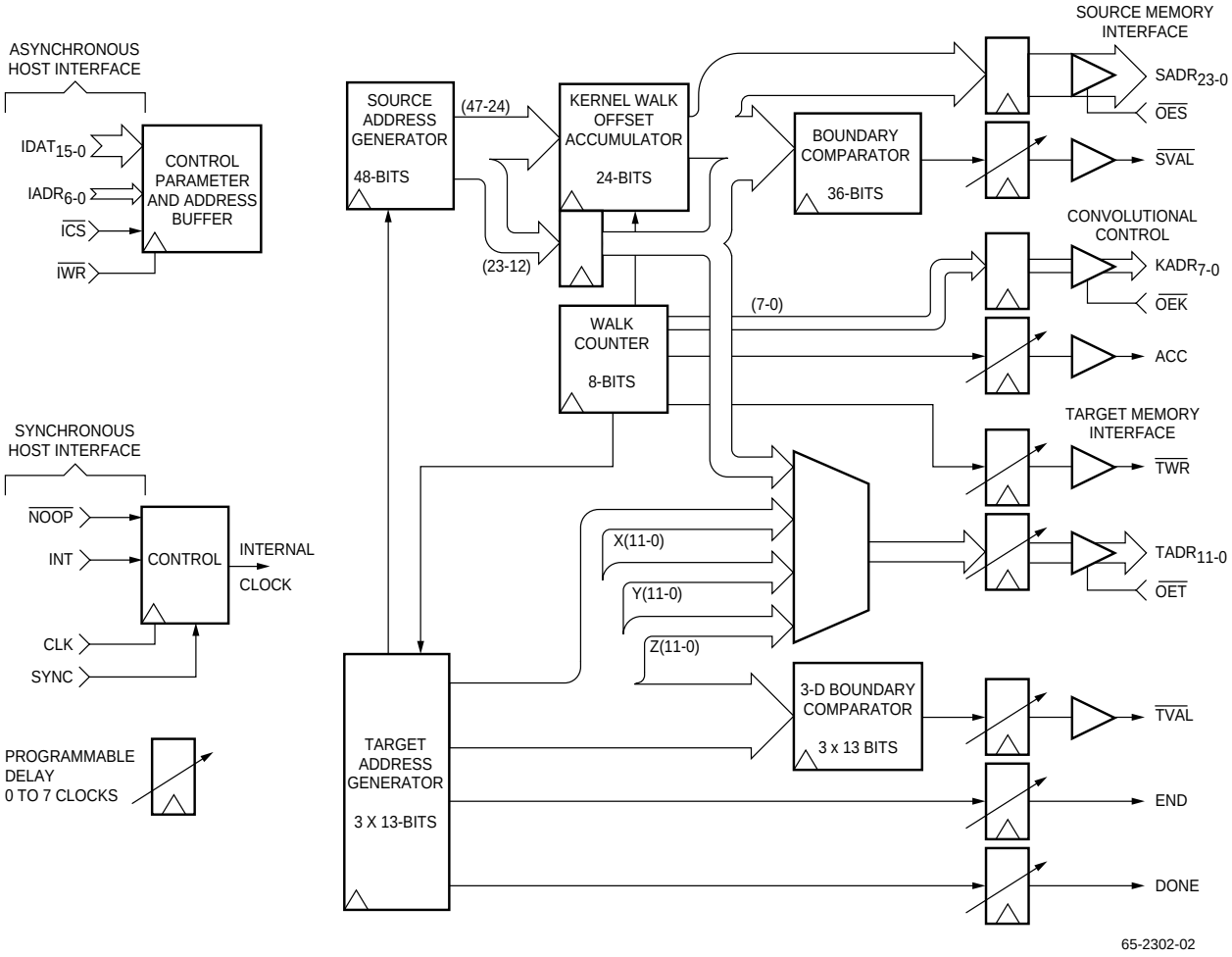
Description (continued)

A system based on two TMC2302As can nearest-neighbor resample a two-dimensional 512 x 512 pixel image in 6.5 milliseconds, translating, rotating, or warping it, depending on the user-selected transformation parameters. A complete bilinear interpolation of the sample image can be completed in 26 milliseconds (or 6.5ms with a TMC2246A companion chip), while a nearest-neighbor resampling of a 3D image 128 pixels on a side takes only 53 milliseconds with three TMC2302As. Image resampling speed is independent of angle of rotation, degree of warp, or amount of zoom specified.

The TMC2302A can process image data fields with up to 24 bits of binary resolution (2^{24} pixels) per dimension, with 0 to 16-bit subpixel resolution.

Along with the original Plastic Pin Grid Array (PPGA) package, the TMC2302A is offered in a 120-lead Metric Quad FlatPack (MQFP) as well. All TMC2302 electrical, functional, and environmental specifications are improved or remain unchanged in the TMC2302A.

Block Diagram



Functional Description

General Information

The TMC2302A is a versatile, high-performance address generator which can control, under user direction, filtering or remapping of two or three-dimensional images by resampling them from one set of Cartesian coordinates (x, y, z) into a new, transformed set (u, v, w). Most applications utilize two identical devices for two-dimensional, or three devices for three-dimensional, image processing. The host CPU initializes the system by loading the input image buffer RAM with the source image pixel data and the TMC2302As with the image transformation and system configuration control parameters. These parameters are loaded by a separate, asynchronous input clock. The IMS-based system then executes the entire transformation as programmed, generating a DONE flag upon completion of the transform. The user can program the chip to repeat the transform continuously or to halt at the end.

The IMSs continuously compute the target bit plane (u, v) or bit space addresses (u, v, w) in typical line-by-line, raster-scan serial sequence. For each output pixel address, they compute the corresponding remapped source image coordinates, each of whose upper 24 bits become the source bit plane addresses (x, y). An additional lower twelve bits are available through the target address port in the optional extended address mode. Source image addresses may be generated at up to 40MHz, with the corresponding target image addresses then appearing at up to (40/k)MHz, where "k" is the size of the interpolation kernel implemented. In the two-IMS system, one TMC2302A computes the horizontal coordinates x and u while the other generates the y and v (vertical) addresses. In a three-dimensional system, one additional IMS would provide the z and w (depth or time) coordinates.

To support a wide range of image transformations, the "row" or x/u device implements a 16-term polynomial of the form:

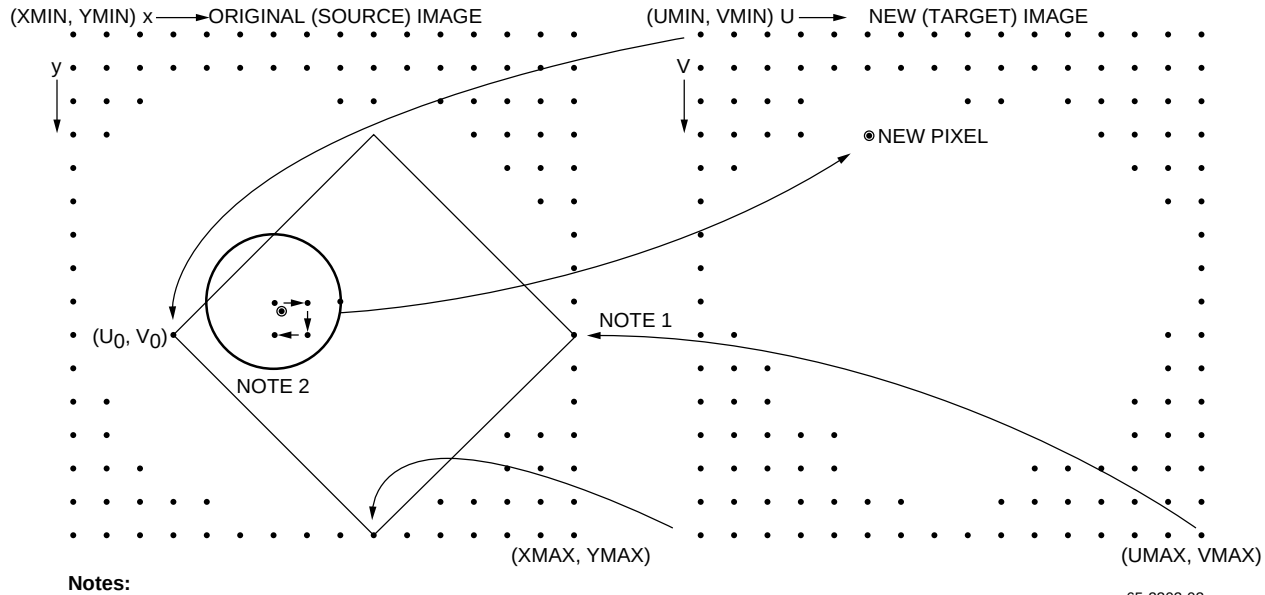
$$x = a + bu + cu^2 + du^3 + ev + fvu + gvu^2 + hvu^3 + iv^2 + jv^2u + kv^2u^2 + lv^2u^3 + mv^3 + nv^3u + ov^3u^2 + pv^3u^3$$

where "a" through "p" are the user-defined image transformation parameters. The TMC2302A steps sequentially through the pixels within a user-defined rectangle in the target image space, computing the "old" source image address (x, y, z) corresponding to each "new" target image pixel (u, v, w). User-programmable flags are available to indicate when the source and target image addresses have fallen outside of a defined rectangular area, simplifying the generation of complex images or image windows. Here, u = U-UMIN and v = V-VMIN, where (u,v) is the target address output by the TMC2302A.

In the three-dimensional mode, the x/u transformation equation is:

$$x = a + bu + ev + kw + fuv + ivw + luw + juvw$$

See "The Image Transformation Polynomial" section of the Applications Discussion.



Notes:
1. Coordinate transformation U, V pixel mapped into X, Y coordinates.
2. Bilinear pixel interpolation walk. New U, V pixel intensity calculated from surrounding X, Y pixel neigborhood.

65-2302-03

Figure 1. Image resampling geometry showing two-dimensional image rotation and expansion

Preliminary Information

The TMC2302A utilizes an external multiplier-accumulator or interpolator, connected to the system clock, to calculate the interpolated pixel value for each color. The products of the original source image pixel values surrounding the remapped pixel location (interpolation kernel) and the appropriate weights stored in the coefficient lookup table are summed. The resulting new interpolated image pixel value is then stored in the corresponding (u, v, w) memory location in the target image memory buffer. Next, the target image address is incremented by one in the "u" direction until UMAX is reached (end of line), when u is reset to UMIN, and the v counter is incremented to give the first pixel location in the next line. The process is repeated, proceeding line-by-line through the image, until VMAX is reached. In the case of three-dimensional images, the IMS system also steps through each page in the image, incrementing in the "w" direction with the completion of each image plane until WMAX is reached, and the transformation is complete.

The Image Manipulation Sequencer can support any nearest-neighbor, bilinear interpolation, or cubic convolution resampling. Interpolation kernels of more than one pixel require an external interpolation coefficient lookup table and multiplier-accumulator or multiple multiplier array. One, two, and three-pass algorithms are supported. For each output point in a typical two-dimensional single-pass static image filter, the TMC2302A implements a spiralling pixel resampling algorithm, "walking" around the resampling neighborhood in two dimensions and generating the appropriate coefficient table addresses to sum up the interpolated pixel value in the external pixel interpolator. At the end of each walk, the TMC2302A will advance one pixel along the output scan line and then execute the walk for that next pixel. When performing multiple-pass interpolation, the TMC2302A system proceeds along only one dimension per pass, which requires dimensionally separable, preferably orthogonal, coefficients.

A basic, two-dimensional TMC2302A-based system is shown in Figure 2. In this typical arrangement, two Image Manipulation Sequencers process the image. The only other components needed beyond the source and target image buffer memories are a multiplier-accumulator or pixel interpolator such as the TMC2246A Image Mixer or TMC2250A Matrix Multiplier, and the Interpolation Coefficient Lookup Table RAM or ROM.

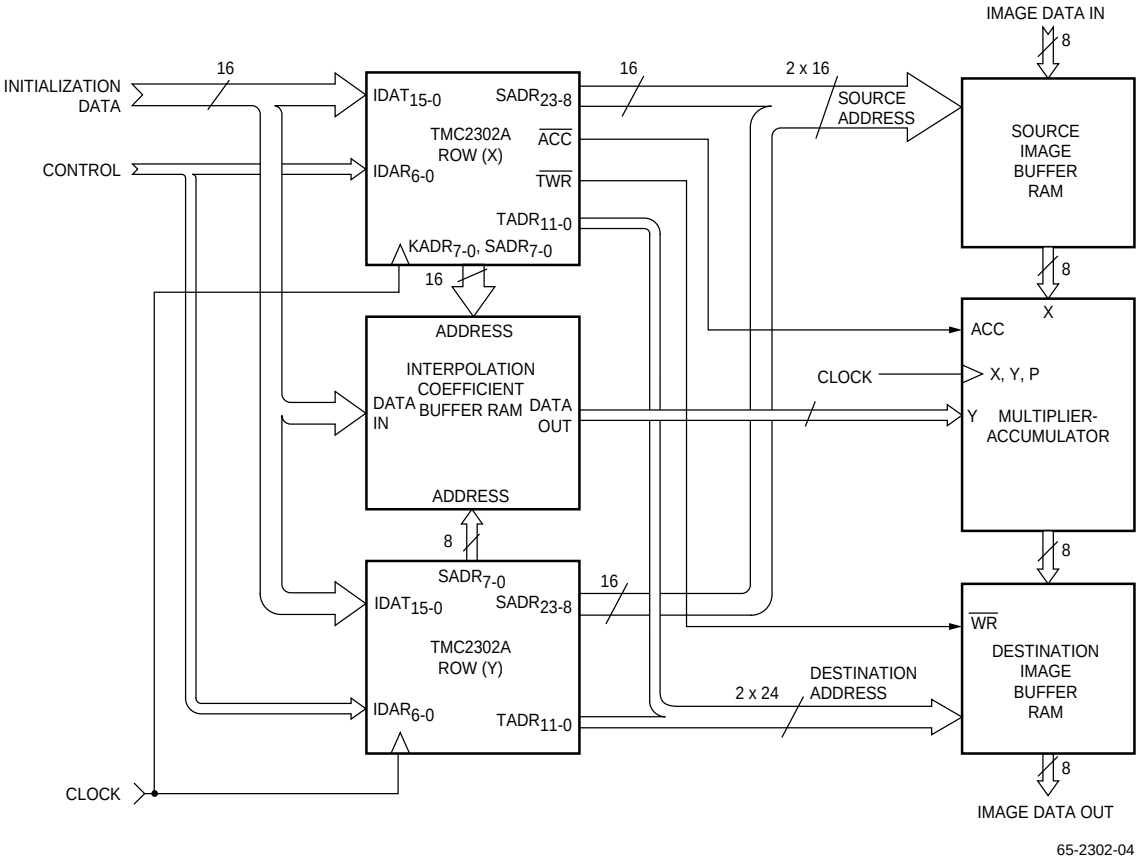
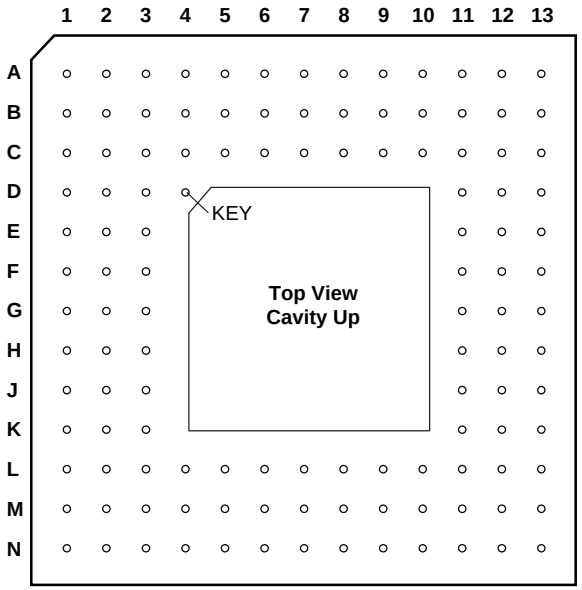


Figure 2. Basic two-dimensional image convolver using TMC2302A IMS with typical 8-bit data path

Pin Assignments

120 Pin Plastic Pin Grid Array, PPGA



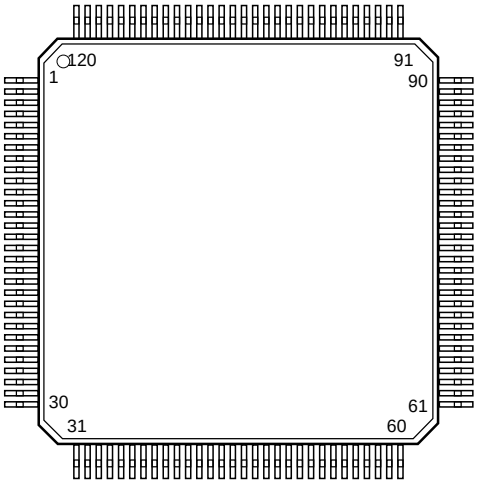
65-2302-05

Pin	Name	Pin	Name	Pin	Name	Pin	Name
A1	GND	C5	$SADR_{19}$	G11	GND	L10	DONE
A2	$SADR_{16}$	C6	$SADR_{22}$	G12	V_{DD}	L11	V_{DD}
A3	$SADR_{17}$	C7	$IADR_5$	G13	$IDAT_0$	L12	GND
A4	V_{DD}	C8	$IADR_1$	H1	$SADR_5$	L13	$\overline{NOOP}$
A5	$SADR_{21}$	C9	$IDAT_{14}$	H2	$SADR_4$	M1	ACC
A6	$\overline{OES}$	C10	$IDAT_{10}$	H3	GND	M2	$\overline{OEK}$
A7	$IADR_6$	C11	GND	H11	GND	M3	$KADR_6$
A8	$IADR_3$	C12	GND	H12	V_{DD}	M4	$KADR_4$
A9	$IADR_0$	C13	$IDAT_6$	H13	SYNC	M5	$KADR_2$
A10	$IDAT_{15}$	D1	$SADR_{11}$	J1	$SADR_3$	M6	$\overline{OET}$
A11	$IDAT_{12}$	D2	$SADR_{12}$	J2	$SADR_2$	M7	$TADR_0$
A12	$IDAT_9$	D3	GND	J3	V_{DD}	M8	$TADR_3$
A13	V_{DD}	D11	V_{DD}	J11	V_{DD}	M9	$TADR_6$
B1	$SADR_{14}$	D12	$IDAT_5$	J12	CLK	M10	$TADR_9$
B2	$SADR_{15}$	D13	$IDAT_4$	J13	$\overline{IWR}$	M11	GND
B3	V_{DD}	E1	$SADR_9$	K1	$SADR_1$	M12	GND
B4	$SADR_{18}$	E2	$SADR_{10}$	K2	$SADR_0$	M13	$\overline{TVAL}$
B5	$SADR_{20}$	E3	GND	K3	GND	N1	GND
B6	$SADR_{23}$	E11	GND	K11	V_{DD}	N2	$KADR_7$
B7	$IADR_4$	E12	$IDAT_3$	K12	INIT	N3	$KADR_5$
B8	$IADR_2$	E13	$IDAT_2$	K13	GND	N4	$KADR_3$
B9	$\overline{ICS}$	F1	$SADR_7$	L1	$\overline{SVAL}$	N5	$KADR_1$
B10	$IDAT_{13}$	F2	$SADR_8$	L2	V_{DD}	N6	$\overline{TWR}$
B11	$IDAT_{11}$	F3	V_{DD}	L3	NC	N7	$TADR_1$
B12	$IDAT_8$	F11	V_{DD}	L4	V_{DD}	N8	$TADR_2$
B13	$IDAT_7$	F12	GND	L5	GND	N9	$TADR_5$
C1	$SADR_{13}$	F13	$IDAT_1$	L6	$KADR_0$	N10	$TADR_7$
C2	V_{DD}	G1	$SADR_6$	L7	V_{DD}	N11	$TADR_{10}$
C3	V_{DD}	G2	GND	L8	$TADR_4$	N12	$TADR_{11}$
C4	GND	G3	V_{DD}	L9	$TADR_8$	N13	ENDD

Preliminary Information

Pin Assignments (continued)

120 Lead Metric Quad Flat Pack, MQFP



65-2302-06

Pin	Name	Pin	Name	Pin	Name	Pin	Name
1	V_{DD}	31	$\overline{OEK}$	61	V_{DD}	91	GND
2	$SADR_{15}$	32	$KADR_7$	62	GND	92	$IDAT_8$
3	$SADR_{14}$	33	V_{DD}	63	$\overline{TVAL}$	93	$IDAT_9$
4	GND	34	$KADR_6$	64	V_{DD}	94	$IDAT_{10}$
5	V_{DD}	35	$KADR_5$	65	GND	95	$IDAT_{11}$
6	$SADR_{13}$	36	$KADR_4$	66	$\overline{NOOP}$	96	$IDAT_{12}$
7	$SADR_{12}$	37	GND	67	INIT	97	$IDAT_{13}$
8	GND	38	$KADR_3$	68	V_{DD}	98	$IDAT_{14}$
9	$SADR_{11}$	39	$KADR_2$	69	GND	99	$IDAT_{15}$
10	$SADR_{10}$	40	$KADR_1$	70	CLK	100	$\overline{ICS}$
11	$SADR_9$	41	$KADR_0$	71	$\overline{IWR}$	101	$IADR_0$
12	V_{DD}	42	$\overline{OET}$	72	GND	102	$IADR_1$
13	$SADR_8$	43	$\overline{TWR}$	73	V_{DD}	103	$IADR_2$
14	$SADR_7$	44	$TADR_0$	74	SYNC	104	$IADR_3$
15	GND	45	V_{DD}	75	V_{DD}	105	$IADR_4$
16	V_{DD}	46	$TADR_1$	76	GND	106	$IADR_5$
17	$SADR_6$	47	$TADR_2$	77	$IDAT_0$	107	$IADR_6$
18	$SADR_5$	48	$TADR_3$	78	$IDAT_1$	108	$\overline{OES}$
19	$SADR_4$	49	$TADR_4$	79	GND	109	$SADR_{23}$
20	GND	50	$TADR_5$	80	V_{DD}	110	$SADR_{22}$
21	$SADR_3$	51	$TADR_6$	81	$IDAT_2$	111	$SADR_{21}$
22	$SADR_2$	52	$TADR_7$	82	$IDAT_3$	112	$SADR_{20}$
23	$SADR_1$	53	$TADR_8$	83	$IDAT_4$	113	V_{DD}
24	V_{DD}	54	$TADR_9$	84	GND	114	$SADR_{19}$
25	$SADR_0$	55	$TADR_{10}$	85	$IDAT_5$	115	$SADR_{18}$
26	$\overline{SVAL}$	56	$TADR_{11}$	86	$IDAT_6$	116	$SADR_{17}$
27	ACC	57	DONE	87	$IDAT_7$	117	$SADR_{16}$
28	GND	58	GND	88	V_{DD}	118	GND
29	V_{DD}	59	NC	89	GND	119	V_{DD}
30	GND	60	ENDD	90	V_{DD}	120	GND

Pin Descriptions

Pin Name	Pin Number PPGA	Pin Number MQFP	Pin Function Description
Power			
V_{DD}	C3, C2, F3, G3, J3, L2, L4, L7, L11, K11, J11, H12, G12, F11, D11, A13, A4, B3	1, 5, 12, 16, 24, 29, 33, 45, 61, 64, 68, 73, 75, 80, 88, 90, 113, 119	**Supply Voltage.** The TMC2302A operates from a single +5V supply. All pins must be connected.
GND	D3, E3, G2, H3, K3, N1, L5, M11, M12, L12, K13, H11, G11, F12, E11, C12, C11, C4, A1	4, 8, 15, 20, 28, 30, 37, 58, 62, 65, 69, 72, 76, 79, 84, 89, 91, 118, 120	**Ground.**

Pin Descriptions (continued)

Pin Name	Pin Number PPGA	Pin Number MQFP	Pin Function Description
Clocks			
CLK	J12	70	**System Clock .** The pixel clock of the TMC2302A strobes all internal registers except the control parameter preload registers. All timing specifications except those are referenced to the rising edge of CLK.
$\overline{IWR}$	J13	71	**Input Parameter Write Clock.** The internal image transformation and configuration control parameter registers are double buffered to simplify interfacing with system controllers. Depending on the state of the chip selects $\overline{ICS}$, control words input to $IDAT_{15-0}$ and the corresponding addresses presented to $IADR_{6-0}$ are strobed into the outer preload registers on the rising edge of the Input parameter Write clock $\overline{IWR}$. The last parameter must be loaded twice on two consecutive rising edges of $\overline{IWR}$.
Inputs			
$IDAT_{15-0}$	A10, C9, B10, A11, B11, C10, A12, B12, B13, C13, D12, D13, E12, E13, F13, G13	99, 98, 97, 96, 95, 94, 93, 92, 87, 86, 85, 83, 82, 81, 78, 77	**Input Parameter Data.** Configuration and transformation parameter Input Data are presented, along with the appropriate input register address word $IADR_{6-0}$, to the parameter Input Data port, and are latched into the preload registers on the next rising edge of $\overline{IWR}$. Preload register updates are disabled by the chip select control $\overline{ICS}$. See Figure 3.
$IADR_{6-0}$	A7, C7, B7, A8, B8, C8, A9	107, 106, 105, 104, 103, 102, 101	**Input Parameter Address.** The input parameter preload register currently indicated by the Input parameter register Address $IADR_{6-0}$ is loaded with the data presented to input port IDAT on the rising edge of $\overline{IWR}$, as demonstrated in Figure 3.
Outputs			
$SADR_{23-0}$	B6, C6, A5, B5, C5, B4, A3, A2, B2, B1, C1, D2, D1, E2, E1, F2, F1, G1. H1, H2, J1, J2, K1, K2	109, 110, 111, 112, 114, 115, 116, 117, 2, 3, 6, 7, 9, 10, 11, 13, 14, 17, 18, 19, 21, 22, 23, 25	**Source Address.** The 24-bit address of one dimension (X, Y, Z) of the source image pixel value currently being resampled is output through the Source Address port $SADR_{23-0}$. This port can be forced to the high-impedance state by the enable control $\overline{OES}$.
$KADR_{7-0}$	N2, M3, N3, M4, N4, M5, N5, L6	32, 34, 35, 36, 38, 39, 40, 41	**Coefficient Address.** The integer address steps for each dimension of the spiral interpolation walk performed by the TMC2302A, as determined by the transform parameter KERNEL, are generated by the internal walk counter and output at the Coefficient Address output port $KADR_{7-0}$. This port can be forced to the high-impedance state by the enable control $\overline{OEK}$.

Preliminary Information

Pin Descriptions (continued)

Pin Name	Pin Number		Pin Function Description
	PPGA	MQFP	
$TADR_{11-0}$	N12, N11, M10, L9, N10, M9, N9, L8, M8, N8, N7, M7	56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 44	**Target Address.** The 12-bit address of one dimension (U, V, W) of the target image pixel value just resampled is output through the Target Address Port $TADR_{11-0}$. This port is forced into the high-impedance state by the enable control $\overline{OET}$. $TADR_{11-0}$ can be delayed up to seven clock cycles after the nominal sequence shown in Table 4 by utilization of the pipeline delay parameter PIPTAD. For systems requiring greater spatial resolution in the source image than that offered by the $SADR_{23-0}$ alone, the Target Address Port can be reconfigured to output 12 additional LSBs of the source address by placing the device into the Extended mode, in which case the pipeline delay parameter must be set to 0 to maintain alignment with the current source address port output. See the Device Configuration and Control Parameters section.
Controls			
INIT	K12	67	**Initialize.** The TMC2302A control logic is cleared and initialized for the start of a new image transformation, and the internal working registers are updated with the contents of the current control parameter preload registers when the registered control input INIT is HIGH. The image transformation then commences with the first source image pixel address nine clocks after INIT is returned low.
SYNC	H13	74	**Run/Halt.** The user can select between continuous or one-frame operation with the registered input control SYNC. Assuming that INIT remains LOW and $\overline{NOOP}$ remains HIGH, if SYNC remains HIGH at the end of a transform the TMC2302A will begin the next image transformation without interruption. This assumes either that the user is not changing the parameter set, or that a new set of parameters has already been loaded into the preload registers midframe, prior to the beginning of the last line in the transform. If SYNC is LOW during the last clock cycle of a transform, the device will complete the image, having loaded the new transform parameter set during the first clock of the final line of the transform, and halt in the state set on the first clock cycle of the next transform. These outputs are held until SYNC is again brought HIGH, and operation resumes on the next clock. See Figure 5.
$\overline{ICS}$	B9	100	**Input Parameter Chip Select.** The input parameter preload register write clock $\overline{IWR}$, and thus the preloading of all configuration and transformation parameters, is disabled on the next clock when the registered Input parameter Chip Select input is HIGH. When $\overline{ICS}$ returns LOW, they are enabled on the next clock. See Figure 3.
ACC	M1	27	**Accumulate.** The external pixel interpolator or multiplier-accumulator is initialized for a new accumulation of products by the registered Accumulator Control output ACC. On the first cycle of each interpolation walk, this output goes LOW for one cycle, effectively clearing the register by loading in only the first new resampled pixel value. When performing nearest-neighbor resampling, this control will remain LOW throughout the entire transform. This output can be delayed up to seven clock cycles after the nominal sequence shown in Table 4 by the pipeline delay parameter PIPACC. See the Device Configuration and Control Parameters section.

Pin Descriptions (continued)

Pin Name	Pin Number		Pin Function Description
	PPGA	MQFP	
$\overline{TWR}$	N6	43	**Target Memory Write Enable.** On the last cycle of each interpolation walk, the Target Write Enable goes LOW for one clock cycle, returning HIGH for all but the last cycle of the next walk. When performing nearest-neighbor resampling, this control will remain LOW throughout the entire transform. This output can be forced to the high-impedance state by the enable control $\overline{OET}$, and can be delayed up to seven clock cycles after the nominal sequence shown in Table 4 by the pipe-line delay parameter PIPTWR. See the Device Configuration and Control Parameters section.
$\overline{NOOP}$	L13	66	**No Operation.** Assuming that INIT remains LOW, the internal system clock of the TMC2302A will be disabled on the next clock, halting the current transform, when the registered control input $\overline{NOOP}$ goes LOW. When $\overline{NOOP}$ returns HIGH, normal operation resumes on the next clock. This control does not affect the loading of the configuration and transformation parameter preread registers.
$\overline{OES}$	A6	108	**Source Address Output Enable.** The source address port $SADR_{23-0}$ is enabled when the asynchronous output enable $\overline{OES}$ is LOW. When $\overline{OES}$ is HIGH, the port is in the high-impedance state.
$\overline{OEK}$	M2	31	**Coefficient Address Output Enable.** The interpolation coefficient address port $KADR_{7-0}$ is enabled when the asynchro- nous output enable $\overline{OEK}$ is LOW. When $\overline{OEK}$ is HIGH, the port is in the high-impedance state.
$\overline{OET}$	M6	42	**Target Address Output Enable.** The target address port $TADR_{11-0}$ and target write enable $\overline{TWR}$ are enabled when the asynchronous Target Output Enable $\overline{OET}$ is LOW. When $\overline{OET}$ is HIGH, these outputs are in the high-impedance state. This control functions in both the normal and extended addressing modes.
Flags			
$\overline{SVAL}$	L1	26	**Source Address Valid.** When the current source image address component output is within the working space defined by the parameters XMIN and XMAX (or YMIN, YMAX for the column (Y/V) device or ZMIN, ZMAX for the page (Z/W) device), the Source Address Valid flag $\overline{SVAL}$ for that device is LOW. This flag will go HIGH on the clock in which the corresponding component address falls outside the defined region. In a typical system, the $\overline{SVAL}$ outputs of all IMS devices are OR'ed together to generate a global boundary violation flag. The user might then insert zeroes into the pixel interpolator to ignore that portion of the image outside the defined space, or insert a background color or image. This output can be delayed up to seven clock cycles after the nominal sequence shown in Table 4 by the pipeline delay parameter PIPSVA. See the Device Configuration and Control Parameters section.

Pin Descriptions (continued)

Pin Name	Pin Number		Pin Function Description
	PPGA	MQFP	
$\overline{\text{TVAL}}$	M13	63	**Target Address Valid.** When the current target image addresses are within the working space defined by the parameters UMINI and UMAXI, and VMINI and VMAXI (and WMINI and WMAXI for systems processing three-dimensional images), the Target Address Valid flag $\overline{\text{TVAL}}$ for that device is LOW. This flag will go HIGH on the clock in which the current target address outputs fall outside the defined region, which must fall inside the target area defined by UMIN, UMAX, etc. Since each TMC2302A device is programmed with distinct MINI/MAXI parameters and generates a separate $\overline{\text{TVAL}}$ flag, the user may define separate two or three-dimensional target space windows for each device. $\overline{\text{TVAL}}$ can be delayed up to seven clock cycles after the nominal sequence shown in Table 4 by the pipeline delay parameter PIPTVA. See the Device Configuration and Control Parameters section.
ENDD	N13	60	**End of Dimension.** During the last pixel interpolation walk of a row (X/U device), the last row in a page (Y/V device), or the last page in a three-dimensional transform (Z/W device), the flag ENDD goes HIGH for the entire walk, indicating End of the transform in that dimension. It remains LOW otherwise. This output can be delayed up to seven clock cycles after the nominal sequence shown in Table 4 by the pipeline delay parameter PIPEND. See the Device Configuration and Control Parameters section.
DONE	L10	57	**Done.** On the last clock cycle of the current image transform, the DONE flags on all TMC2302As go HIGH for one clock cycle. On the next clock cycle, all devices output the first addresses and control signals for the next image transform. If SYNC is LOW, the IMS system halts. If SYNC is HIGH, operation continues without interruption. See “SYNC,” in the Controls section. This flag can be delayed up to seven clock cycles after the nominal sequence shown in Table 4 by the pipeline delay parameter PIPDON. Also see “PFLS,” in the Device Configuration and Control Parameters section.
No Connects			
NC	L3	59	**No Connect.**
	D4	—	**Index Pin.**

Transformation Coefficient and Configuration and Control Parameters

The TMC2302A is intended to act as a co-processor, requiring only that the user program the device to perform the image transformation desired by loading in the appropriate device configuration and transformation control parameters discussed in this section. The user then issues an “Init” command, allowing his system to run unattended until the completion of the image when a “Done” flag is generated to inform the host system.

The capabilities and flexibility of the TMC2302A Image Manipulation Sequencer are apparent when reviewing the following tables which define the transformation coefficient and configuration and control parameters. These tables are broken up into two separate groups. The first parameters discussed are the control words which select the dimension calculated, the functional configuration of each device, the working space in which they will operate, the size of the interpolation kernel desired, and the timing of the various address and control signals involved in handling the pixel data pipeline. The second parameters are the polynomial transform coefficients used in performing image manipulation. The TMC2302A utilizes three levels of internal 48-bit accumulators to calculate these values by forward difference accumulation, generating no significant cumulative spatial error for most applications. The user must be aware that all internal parameter and coefficient registers must be set by the user, including resetting after powerup any unused control words or coefficients.

As mentioned above, the TMC2302A also features user-programmable image data pipeline configuration controls. All output signals except the source and coefficient address outputs can be individually delayed by the user up to seven clocks after the nominal system timing illustrated in Table 4. This allows the user to software-configure the TMC2302As in his system to match his pixel interpolator, image buffer, and interpolation coefficient RAM structure timing.

The user can also program the device to continue into the next image for a set number of clock cycles after the Done flag has appeared. First, this “flushes” the final resampled pixel data word through the interpolation pipeline, all the way to the target image RAM. Also, valid pixel data will then appear on the first clock of the next transform independent of the length of the pixel pipeline, incurring no lost clock cycles.

Device Configuration and Control Parameters

UMIN, VMIN, WMIN — The memory addresses of the target image boundaries corresponding to the top, left side, and front page of the new image being generated are defined in all devices of the user's system by the parameters UMIN, VMIN, and WMIN, respectively. At the beginning of the transformation, the initial source image coordinate (X_0, Y_0, Z_0) will be mapped to this coordinate set. The numeric format assumed is 12-bit unsigned binary integer.

UMAX, VMAX, WMAX — The memory addresses of the target image boundaries corresponding to the bottom, right side, and last page of the image being generated are defined in all devices by the parameters UMAX, VMAX, and WMAX, respectively. These values should be greater than the UMIN/VMIN/WMIN values defined above. Numeric format assumed is unsigned 12-bit binary integer.

Note: The parameter UMAX must exceed UMIN so as to ensure that a minimum of 5 system clock cycles in two-dimensional operation, or 15 clock cycles in three-dimensional operation, pass between the periods in which these two target address values are generated. Thus in 2D nearest neighbor operation UMAX must be 5 greater than UMIN. In 2D bilinear interpolation mode (4-pixel two-dimensional kernel) the distance must be two pixels in the target image (actually enforcing a spacing of 8 system clocks).

UMINI, VMINI, WMINI — The target image addresses corresponding to those of the top, left side, and front page of the 2 or 3 dimensional region indicated by the valid target address flag $\overline{\text{TVAL}}$ are UMINI, VMINI, and WMINI, respectively. Thus, to define a valid region beginning at “m,” the MINI parameter value is “m,” These parameters are assumed to be in 12-bit unsigned binary integer format. Proper $\overline{\text{TVAL}}$ operation requires UMIN < UMINI < UMAXI < UMAX, etc.

UMAXI, VMAXI, WMAXI — The target image addresses one more than those of the right side, bottom and back page of the region indicated by the valid target address flag $\overline{\text{TVAL}}$ are UMAXI, VMAXI, and WMAXI, respectively. Thus, to define a valid region ending at “n,” the MAXI parameter value is “n+1”. These parameters are assumed to be in 12-bit unsigned integer format.

Preliminary Information

XMIN, XMAX — The source image boundaries are defined for each device by the parameters XMIN and XMAX, in the case of the row device. The column device then contains YMIN and YMAX, and the page device (in systems performing three-dimensional operations) ZMIN and ZMAX. The value of XMAX should be greater than XMIN if the boundary violation flag $\overline{SVAL}$ is to operate correctly. These values are assumed to be in 32-bit unsigned binary integer format.

PFLS — The user can set the number of clock cycles that the TMC2302A continues in to the next image following the DONE flag, allowing his system to Flush all control and data pipeline paths and halt after a maximum of seven cycles. The numeric format assumed is three-bit unsigned binary integer.

PTAD, PDON, PEND, PTVA, PSVA, PTWR, PACC — As mentioned above, the control signals and target image pixel addresses generated by the TMC2302A can be delayed up to seven clock cycles after the nominal timing shown in Table 4 by setting the appropriate Pipeline delay word. The numeric format assumed for all delay words is three-bit unsigned binary integer.

XTND — When the user sets the control bit XTND to 1, the TMC2302A operates in an extended-resolution source address bus configuration. Assuming that the user has his own raster scan generator available elsewhere to manage the flow of output pixels from the TMC2302A system, the target address output bus $TADR_{11-0}$ is reconfigured internally into an extension of the source address bus, as $SADR_{11-0}$. The original source address bus $SADR_{23-0}$ is then $SADR_{35-12}$, providing 36 bits of spatial resolution in the source address space. An XTND of 0 puts the device in the standard 24-bit source, 12-bit target address configuration.

E3D — Setting this control bit to 0 indicates a two-dimensional image transform is to be performed. When the E3D is set to 1, a three-dimensional image is assumed, using three TMC2302A devices.

DIM — The user sets each TMC2302A to operate in a specific dimension as follows:

$DIM_{1,0}$	Dimension
00	X/U (Row) Device
01	Y/V (Column) Device
10	Z/W (Page) Device
11	No Operation

MODE — In systems performing the standard two-dimensional spiral interpolation walk, MODE is set to 11, indicating single-pass operation. When performing multiple-pass resampling, the user must set this two-bit control word pass-by-pass in all IMSs, to implement each pass direction. For instance, setting MODE to 00 causes the TMC2302A system to increment only in the X-direction, holding the Y (and Z) addresses constant until the end of that pixel walk. On the next pass through the image, the user sets MODE = 01, with the kernel increment in Y only. In 3D, the IMS system then proceeds again through the (U, V) target image space, walking kernels only along the Z direction.

$Mode_{1,0}$	Resampling Performed
00	X-Pass
01	Y-Pass
10	Z-Pass
11	Two-Dimension Spiral Walk

KERNEL — This parameter determines the size of the interpolation walk performed. To implement a convolutional sum of K+1 pixels, the parameter KERNEL is set to K, up to a maximum of 255. In single-pass operation, this value must be identical in all devices, giving a square interpolation kernel. In multiple-pass operation, however, non- square kernels may be implemented, with different K values in each dimension. Or, the user could utilize a banded memory architecture in two-pass mode to access an entire row or column of a kernel in one clock, completing the entire sum in a single pass through the other dimension of the kernel. Numeric format is 8-bit unsigned integer.

FOV — The user determines the size of each step in an interpolation walk, in terms of the number of source image pixels, by setting the Field Of View control. The binary weighting of the image transformation parameters and source address must be taken into account when determining this value. See Table 6 and the Applications Discussion section. The numeric format assumed is unsigned 16-bit integer.

Table 1. Control Parameter Registers Binary Format (Row, Column or Page Device)

Addr		Format																Limits	
Name	Hex	MSB															LSB	Dec	Hex
UMIN	30					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
UMAX	31					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
UMINI	32					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
UMAXI	33					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
VMIN	34					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
VMAX	35					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
VMINI	36					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
VMAXI	37					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
WMIN	38					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
WMAX	39					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
WMINI	3A					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
WMAXI	3B					2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	4095 0	FFF 000
XMINL	3C	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	0	00000000
XMINM	3D	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	2^{32}-1	FFFFFFFF
XMAXL	3E	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	0	00000000
XMAXM	3F	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	2^{32}-1	FFFFFFFF
PFLS	40		2^{2}	2^{1}	2^{0}													7 0	7 0
PTAD	40					2^{2}	2^{1}	2^{0}										7 0	7 0
PDON	40								2^{2}	2^{1}	2^{0}							7 0	7 0
PEND	40											2^{2}	2^{1}	2^{0}				7 0	7 0
PTVA	40														2^{2}	2^{1}	2^{0}	7 0	7 0
PSVA	41		2^{2}	2^{1}	2^{0}													7 0	7 0
PTWR	41					2^{2}	2^{1}	2^{0}										7 0	7 0
PACC	41								2^{2}	2^{1}	2^{0}							7 0	7 0
XTND	41											XTND							
E3D	41												E3D						
DIM	41													DIM_1	DIM_0				
MODE	41															$MODE_1$	$MODE_0$		
KERNEL	42									2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	255 0	FF 00
FOV	43	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	2^{16}-1 0	FFFF 0000

Preliminary Information

Transformation Parameter Registers

The Transformation Parameter Word storage register addresses for the X/U device are listed in Table 2, along with the differential terms for each polynomial coefficient for both two and three-dimensional transforms. The polynomial terms for the other IMS device(s) are found by replacing every "X" in the table with a Y (or Z). A TMC2302A-based system can perform image manipulations of up to third order in two dimensions, and three-dimensional transforms of up to order 1.5 ("first-and-a-half order"). Also, see "The Image Transformation Polynomial", in the Applications Discussion section.

The notation used to define each polynomial coefficient term in Table 2 is easily interpreted. Each differential is of course defined by a differential in X, followed by the corresponding dependent U, V, or W terms. Thus,

DXUV is equivalent to $d^2X/dUdV$

and DXUUUV to d^4X/dU^3dV.

Table 2. Transformation Polynomial Coefficient Register Addresses

Name	Parameter		Coefficient Word Addresses (hex)		
	2D Term	3D Term	MSW	CSW	LSW
A	X_0	X_0	00	01	02
B	DXU	DXU	03	04	05
C	DXUU		06	07	08
D	DXUUU		09	0A	0B
E	DXV	DXV	0C	0D	0E
F	DXUV	DXUV	0F	10	11
G	DXUUV	X_0	12	13	14
H	DXUUUV	DXU	15	16	17
I	DXVV	DXVW	18	19	1A
J	DXUVV	DXUVW	1B	1C	1D
K	DXUUVV	DXW	1E	1F	20
L	DXUUUVV	DXUW	21	22	23
M	DXVVV		24	25	26
N	DXUVVV		27	28	29
O	DXUUVVV		2A	2B	2C
P	DXUUUVVV		2D	2E	2F

Note:

1. The X_0 and DXU terms must each be loaded into two different registers when performing 3D transforms. Table 2 shows the binary weighting of all of the Transformation Parameter words, which are 48-bit signed fractional binary.

Table 3. Integer Binary Weighting of Transformation Parameters

	Format																Limits	
	MSB															LSB	Dec	Hex
MSB	-2^{47}	2^{46}	2^{45}	2^{44}	2^{43}	2^{42}	2^{41}	2^{40}	2^{39}	2^{38}	2^{37}	2^{36}	2^{35}	2^{34}	2^{33}	2^{32}	$2^{48}-1$	FFFFFFFFFFFF
CSW	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}		
LSW	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	0	000000000000

Note:

1. A minus sign indicates a sign bit.

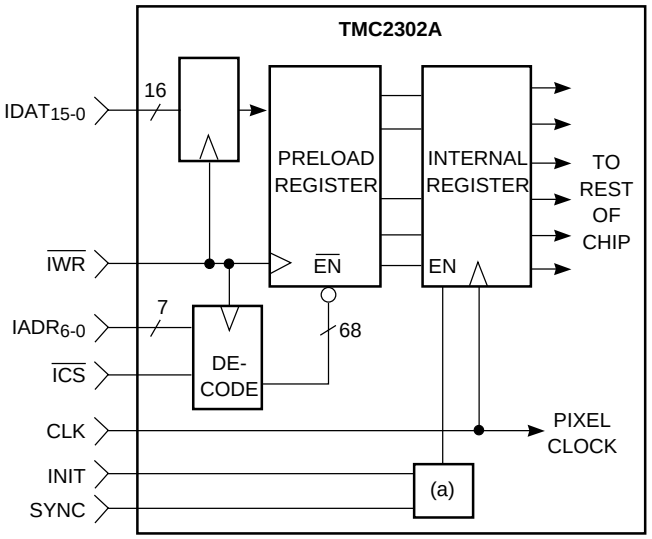


Figure 3. Image transformation and configuration control parameters register structure

Figure 3 depicts the control preload register structure and Figure 4B gives the corresponding timing relationships.

Table 4. Nominal Output Signal Timing

$SADR_{23-0}$[1]	ACC	$TADR_{11-0}$	$\overline{TWR}$	END	DONE
$X_{I-1,J,0}$	0	$U_{L-1,M}$	1	0	0
$X_{I-1,J,1}$	1	$U_{L-1,M}$	1	0	0
$X_{I-1,J,2}$	1	$U_{L-1,M}$	1	0	0
•					
•					
•					
$X_{I-1,J,K}$	1	$U_{L-1,M}$	0	1	0
$X_{I,J,0}$	0	$U_{L,M}$	1	1	0
$X_{I,J,1}$	1	$U_{L,M}$	1	1	0
$X_{I,J,2}$	1	$U_{L,M}$	1	1	0
•					
•					
•					
$X_{I,J,K}$	1	$U_{L,M}$	0	1	1

Note:
1. $KADR_{7-0}$ timing identical.

The nominal sequence of address and control signals of a two-dimensional, single-pass-programmed TMC2302A system, with all PIPE parameters set to 0, is shown in Table 4. Here, the values of the last two new target image pixels $U_{L-l,M}$ and $U_{L,M}$ are being calculated, and the beginning and end of the interpolation walks of length K which sample source image pixels in the neighborhod of locations ($X_{I-1,J}$, $X_{I,J}$) can be seen. Utilizing the arrival of the source image address ($SADR_{31-0}$) as a reference point, the other signals shown can be delayed up to seven clock cycles from the nominal timing shown here, allowing the user to configure these outputs to match the timing latencies of his pixel data path structure. Considerable speed and timing variations in image buffer memory, data register, and pixel interpolator structure can thus be accomodated, with minimal corresponding support hardware. Also see “PFLS,” in the Device Configuration and Control Parameters section.

Equivalent Circuits and Threshold Levels

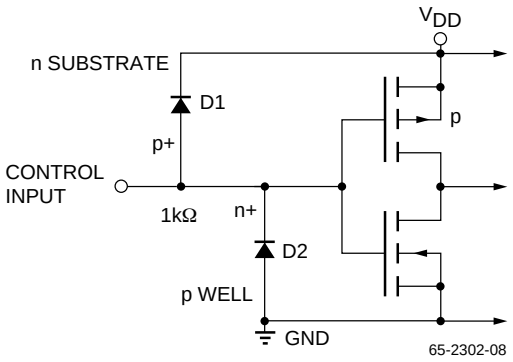


Figure 5. Equivalent Input Circuit

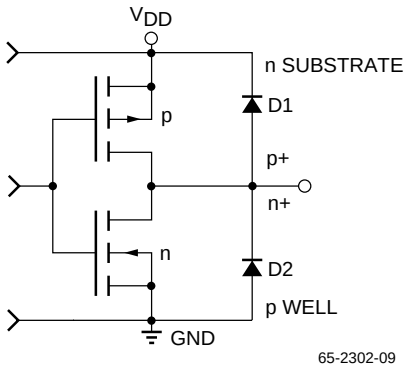


Figure 6. Equivalent Output Circuit

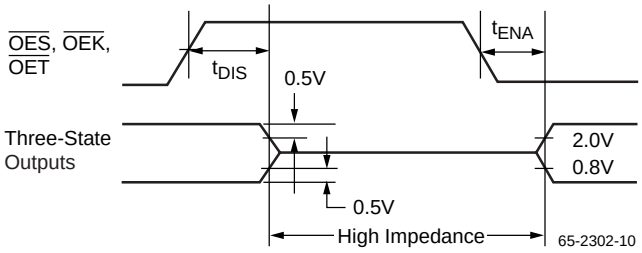


Figure 7. Threshold Levels for Three-State Measurements

Absolute Maximum Ratings

(beyond which the device may be damaged)[1]

Parameter	Min.	Max.	Units
Supply Voltage	-0.5	+ 7.0	V
Input Voltage	-0.5	V_{DD} + 0.5	V
Output applied voltage	-0.5	V_{DD} + 0.5	V
Short-circuit duration (single output in HIGH state to ground)		1	second
Operating, case temperature	-60	130°	C
Junction temperature		175°	C
Lead, soldering temperature (10 seconds)		300°	
Storage temperature	-65	+150°	C

Notes:

1. Absolute maximum ratings are limiting values applied individually while all other parameters are within specified operating conditions. Functional operation under any of these conditions is NOT implied.

Preliminary Information

Operating Conditions

Parameter		Test Conditions				-1			Units
			Min.	Nom.	Max.	Min.	Nom.	Max.	
V_{DD}	Supply Voltage		4.75	5.0	5.25	4.75	5.0	5.5	V
V_{IL}	Input Voltage, Logic LOW				0.8			0.8	V
V_{IH}	Input Voltage, Logic HIGH		2.0			2.0			V
I_{OL}	Output Current, Logic LOW				8.0			8.0	mA
I_{OH}	Output Current, Logic HIGH				-4.0			-4.0	mA
t_{CY}	Cycle Time	V_{DD} = Min	33			25			ns
t_{PWL}	Clock Pulse Width, LOW	V_{DD} = Min	15			12.5			ns
t_{PWH}	Clock Pulse Width, HIGH	V_{DD} = Min	15			10			ns
t_S	Input Setup Time		10			8			ns
t_H	Input Hold Time		2			2			ns
T_A	Ambient Temperature, Still Air		0		70	0		70	°C

Electrical Characteristics[1]

Parameter		Test Conditions	Min.	Max.	Units
I_{DDQ}	Supply Current Quiescent	V_{DD} = Max, V_{IN} = 0V		10	mA
I_{DDU}	Supply Current, Unloaded	V_{DD} = Max, f = 20MHz, $\overline{OES}$ = $\overline{OEK}$ = $\overline{OET}$ = 5V		70	mA
I_{IL}	Input Current, Logic LOW	V_{DD} = Max, V_{IN} = 0V	-10		μA
I_{IH}	Input Current, Logic HIGH	V_{DD} = Max, V_{IN} = V_{DD}		10	μA
V_{OL}	Output Voltage, Logic LOW	V_{DD} = Min, I_{OL} = Max		0.4	V
V_{OH}	Output Voltage, Logic HIGH	V_{DD} = Min, I_{OH} = Max	2.4		V
I_{OZL}	High-Z Output Leakage Current, Output LOW	V_{DD} = Max, V_{IN} = 0V	-40		μA
I_{OZH}	Hi-Z Output Leakage Current, Output HIGH	V_{DD} = Max, V_{IN} = V_{DD}		40	μA
I_{OS}	Short-Circuit Output Current	V_{DD} = Max, Output HIGH, one pin to ground, one second duration max.	-20	-70	mA
C_I	Input Capacitance	T_A = 25°C, f = 1MHz		10	pF
C_O	Output Capacitance	T_A = 25°C, f = 1MHz		10	pF

Note:

1. Actual test conditions may vary from those shown, but guarantee operation as specified.

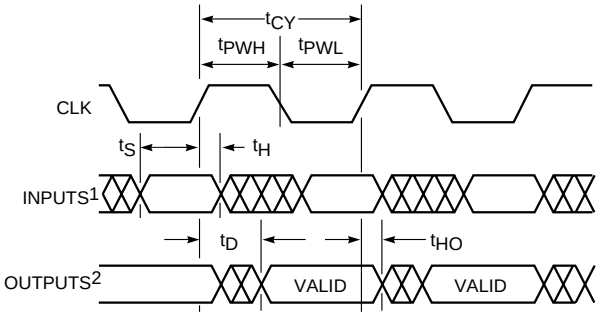
Preliminary Information

Switching Characteristics

Parameter		Test Conditions			-1		Units
			Min.	Max.	Min.	Max.	
t_{DO}	Output Delay	V_{DD} = Min, C_{LOAD} = 25pF		15		12	ns
t_{HO}	Output Hold Time	V_{DD} = Max, C_{LOAD} = 25pF	4		4		ns
t_{ENA}	Three-State Output Enable Delay[1]	V_{DD} = Min, C_{LOAD} = 25pF		12		12	ns
t_{DIS}	Three-State Output Disable Delay[1]	V_{DD} = Min, C_{LOAD} = 25pF		15		15	ns

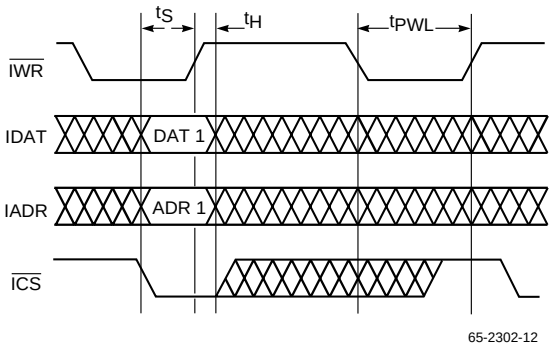
Note:
1. All transitions are measured at a 1.5V level except for t_{DIS} and T_{EMA}.

Timing Diagrams



Notes:
1. Except $\overline{OES}$, $\overline{OET}$, and $\overline{OEK}$.
2. Assumes $\overline{OES}$, $\overline{OET}$, and $\overline{OEK}$ = LOW. All pipeline latency parameters set to 0.

Figure 4a. Timing Diagram, Pixel Clock, Control, and Outputs



Value "DAT 1" is loaded into address "ADR 1" on the second rising edge of $\overline{IWR}$, since $\overline{ICS}$ = 0, having been acquired by the input register on the first edge.

Figure 4b. Timing Diagram, Preload Parameters

Applications Discussion

The Image Transformation Polynomial

On any given clock cycle, when performing a two-dimensional geometric transformation the addresses output by the row (X/U) TMC2302A are generated by forward difference accumulation according to the following third-order polynomial:

$$x(u,v) = a + bu + cu^2 + du^3 + ev + fvu + gvu^2 + hvu^3 + iv^2 + jv^2 u + kv^2 u^2 + Iv^2 u^3 + mv^3 + nv^3 u + ov^3 u^2 + pv^3 u^3 + FOV \bullet CAX(ca)$$

The polynomial utilized for three-dimensional transforms is:

$$x(u,v,w) = a + bu + ev + kw + fuv + ivw + luw + juvw + FOV \bullet CAX\ (ca)$$

where $0 \le u \le$ UMAX-UMIN, $0 \le v \le$ VMAX-VMIN, $0 \le w \le$ WMAX-WMIN, and the polynomials for the column or page devices are obtained by replacing the x by a y or z, as appropriate.

Preliminary Information

FOV is the 16-bit field-of-view parameter, normally set so that the spiral walk proceeds in single-pixel steps. FOV can be increased to expand the step size and thus the spiral walk, subsampling the image. See Table 1 and Table 6. Also, CAX(ca) is the current value of the coefficient address. See the Interpolation Coefficient Lookup Table Addressing. If the spiral walk isn't used, CAX = 0 and FOV is ignored.

We can reform the two-dimensional polynomial as:

$x(u,v) = (a + ev + iv^2 + mv^3) + (b + fv + jv^2 + nv^3)u + (c + gv + kv^2 + ov^3)u^2 + (d + hv + Iv^2 + pv^3)u^3,$

and retain the simpler three-dimensional form:

$x(u, v, w) = a + bu + ev + kw + fuv + ivw + luw + juvw$

and define each of the polynomial coefficients in arithmetic terms as shown in Table 5.

Table 5. Transformation Polynomial Coefficients

	Parameter			
	Two-Dimensional		Three-Dimensional	
Name	Term	Coefficient	Term	Coefficient
A	X_0	a	X_0	a
B	DXU	b + c + d	DXU	b
C	DXUU	2c + 6d	—	0
D	DXUUU	6d	—	0
E	DXV	e + i + m	DXV	e
F	DXUV	f + g + h + j + k + l + n + o + p	DXUV	f
G	DXUUV	2(g + k + o) + 6(h + l +p)	X_0	a
H	DXUUUV	6(h + l + p)	DXU	b
I	DXVV	2i + 6m	DXVW	i
J	DXUVV	2(j + k + l) + 6(n + o + p)	DXUVW	j
K	DXUUVV	4k + 12l + 12o + 36p	DXW	k
L	DXUUUVV	12l + 36p	DXUW	l
M	DXVVV	6m	—	0
N	DXUVVV	6(n + o + p)	—	0
O	DXUUVVV	12o + 36p	—	0
P	DXUUUVVV	36p	—	0

Understanding the Polynomial Coefficients

An Overview

As the formulae indicate, the source address is a polynomial function of the two (or three) dimensions of the target address. Each of the 16 terms of the equation is of the form:

$$\frac{d^{m+n+p}x}{du^m dv^n dw^p}$$

and may be treated approximately as a mixed partial difference of order m, n, and p.

The simplest term, X_0, is a zeroeth (non-) function of the target addresses; it specifies the source address point corresponding to the upper left point in the target space. X_0 generates image translation or "pan."

The next-simplest terms, dX/dU and dY/dV, govern the relative scales of the source and target images, i.e., how large a step in source space corresponds to a unit step in the corresponding direction in the target space. As long as the cross-terms, dX/dV and dY/dU, are zero, this is a straight scale ("zoom") operation, without rotation or shear.

The first-order cross terms, dX/dV and dY/dU, generate source space displacements perpendicular to unit displacements in the target space, thereby causing shearing of the image. In conjunction with the parallel source terms described above, they govern rotation, shear, and scaling of the image.

Although the actions of the higher-order terms become progressively difficult to describe, all terms behave essentially as partial differences of various orders, and a little thought and common sense will generally lead the user to the proper conclusions. For example, the term dXUU (using the notation of Table 2) is a horizontal scale factor which increases as one progresses across each row, causing a quadratic horizontal warp. In fact, all terms of the form d^mx/du^m or d^ny/dv^n cause only stretching of the image, never rotation.

Interpolation Coefficient Lookup Table Addressing

The external coefficient lookup table RAM stores the interpolation coefficient values used to calculate the value of the new pixel. These values are selected by the user, allowing maximum filtering flexibility. In simple filtering applications, the source and target pixel addresses map one-to-one, and only one interpolation coefficient set is required. These integer addresses are generated for each dimension by the internal walk counters of each TMC2302A.

However, applications performing a coordinate transformation will almost always generate non-integer source pixel addresses; that is, the U (or V) locations will not map to the X (or Y) addresses exactly, and a fractional source address components are generated. The user must then expand the interpolation coefficient lookup table to include spatially-corrected values, as detemnined by the subpixel resolution of the system.

The TMC2301 Image Resampling Sequencer allows the user to trade subpixel resolution against interpolation step size by obtaining the interpolation coefficient addresses directly from the fractional part of the source address. The TMC2302A gives the user 16 different interpolation bit weighting positions. The complete Interpolation Coefficient Address for that dimension then consists of both the 8-bit interpolation walk address $KADR_{7-0}$, weighted to match the source address binary point by the parameter FOV, and the fractional portion of the source pixel address $SADR_{23-0}$, to the desired subpixel resolution. See Table 6.

Internal and External Data Formats

The source address value output by the TMC2302A is a 24-bit two's complement number, with binary point assignable by the user anywhere in the 16 lower bits. The Extended mode appends 12 additional fractional bits for greater output precision. All internal computations include these 24 plus 12 bits, plus an additional 12 lower bits, for 48-bit precision. See Table 6.

Intemally, each TMC2302A's source address (X, Y,or Z) generator computes a 48-bit address through a mode-specific accumulation of the sixteen 48-bit user-specified resampling parameters. The 24 most significant bits of the final accumulation emerge via the source address port whereas the "extend" mode makes the 12 next most-significant bits available at the target address port. The 12 least significant bits are truncated internally.

Source Address Bit Weighting and Setting the Binary Point

When performing nearest-neighbor resampling, the user may arbitrarily trade source image size against subpixel resolution merely by adhering to a single binary point position for all resampling parameters. For example, if the binary point follows the 16 most significant bits in each resampling parameter, then it will appear following the source address’16 most significant bits, leaving 8 (20 in extended mode) bits of subpixel resolution on $SADR_n$.

Since the TMC2302A has no internal limiter, the user should select the source address weighting appropriately. Moving the source address connections to the right and reducing the resampling parameters accordingly, reduces the chance of arithmetic overflow while increasing arithmetic round-error.

In any filtering or resampling operation performing an interpolation walk, the user should set the Field or View (FOV) parameter according to the desired binary point position determined above, as follows. To provide 2^{24} integral pixel positions per dimension, with no subpixel resolution, set FOV = 001 (hex). For 2^{23} positions with 1-bit (0,5) subpixel resolution, FOV = 0010 (hex). Similarly, for 2^9 positions and 15-bit subpixel resolution. FOV = 8000 (hex). As shown in Table 6, using the parameter FOV the user effectively “shifts” the bit weight of the coefficient address word $KADR_{7-0}$ to match the established location of his source address binary point. In each case, the EXTEND mode provides 12 additional bits of subpixel resolution but eliminates the separate target or raster address, which must then be generated elsewhere in the user's system.

Table 6. Relative Bit Weighting – Source Address

Weight Word	2^{47} 2^{46} … 2^{40}	2^{39} … 2^{32}	2^{31} … 2^{25} 2^{24}	2^{23} … 2^{16}	2^{15} … 2^{12} … 2^{8}	2^{7} … 2^{0}
Transform Parameters	-47 46 … 40	39 … 32	31 … 25 24	23 … 16		7 … 0
Internal Source Address Generator	-47 46 … 40	39 … 32	31 … 25 24	23 … 16		7 … 0
Source Address Output $SADR_{23-0}$	-23 22 … 16	15 … 8				
Extended Mode Only $TADR_{11-0}$				11 … 4	3 … 0	
$KADR_{7-0}$						
FOV = 0001			2^{7} … 2^{1} 2^{0}			
FOV = 0002		2^{7}	2^{6} … 2^{0}			
• • •						
FOV = 8000	2^{7} … 2^{1}	2^{0}				

Note:
1. A minus sign indicates a sign bit.

Utilization of the Image Boundary Flags $\overline{SVAL}$ and $\overline{TVAL}$

As mentioned above, the TMC2302A provides two programmable valid address, or boundary flags. The source valid flag $\overline{SVAL}$ is asserted when the current source image address output for that device's source image dimension is within the space defined by the configuration parameters XMIN and XMAX, or YMIN and YMAX, or ZMIN and ZMAX, as appropriate. Also, the target valid flag $\overline{TVAL}$ is available to indicate when the current target image address values fall within the space defined by the configuration parameters UMINI, UMAXI, VMINI, VMAXI, and also WMINI and WMAXI in three-dimensional systems. Note that all of these parameters are each programmed into each individual TMC2302A. Thus, the user could define two (or three) different working spaces, one indicated by each IMS device.

Figure 8 may help clarify the relationships among (X_0, Y_0, Z_0), (UMIN, VMIN, WMIN), and (UMAX, VMAX, WMAX), for the two-dimensional case. With positive first derivatives, (X_0, Y_0) and (UMIN, VMIN) represent the upper left corners of the original image and the new destination field, respectively. The lower right corner of the new transformed image is located at (UMAX, VMAX); the location of the corresponding corner of the original image depends on the values of the derivatives.

Not to be confused with (X_0, Y_0), the points (XMIN, YMIN) and (XMAX, YMAX) define the "usable" rectangular portion of the original image which is indicated by the valid address flag $\overline{SVAL}$; points (X, Y) lying outside this region are ignored in most resampling and filtering applications.

Specifically, the point (X_0, Y_0) is the location from which the TMC2302A system begins the image resampling sequence. Every step beyond that point in the source image space is defined by the address generators implementing the image transformation polynomials.

The valid source address flag feature permits one to construct a mosaic of several abutting subimages in the (X, Y) plane, without danger of edge effect interference between adjacent subimages. Note in the figure that the upper right corner of the resampled source image lies outside the admissible region; in practice, the values fetched at these locations will not be included in the convolutional sums. One might, for instance, program these boundary values to alert the system that an edge is being approached and to modify the interpolation coefficients appropriately, or simply to ignore pixel values outside the defined space.

The $\overline{TVAL}$ however is utilized somewhat differently. Working in unison with the target address working space defined by UMIN/UMAX, etc. the target address valid flag could be programmed to delineate image areas other than the immediate working space, and the flag of each TMC2302A to indicate the unique regions anywhere within the target image. With this flexibility, the user can generate windows, "picture-in-picture" composite multiple images, or simply switch to a background image or border color. To make $\overline{TVAL}$ function properly, the used must set UMIN < UMINI < UMAXI < UMAX; likewise for V and, if used, W.

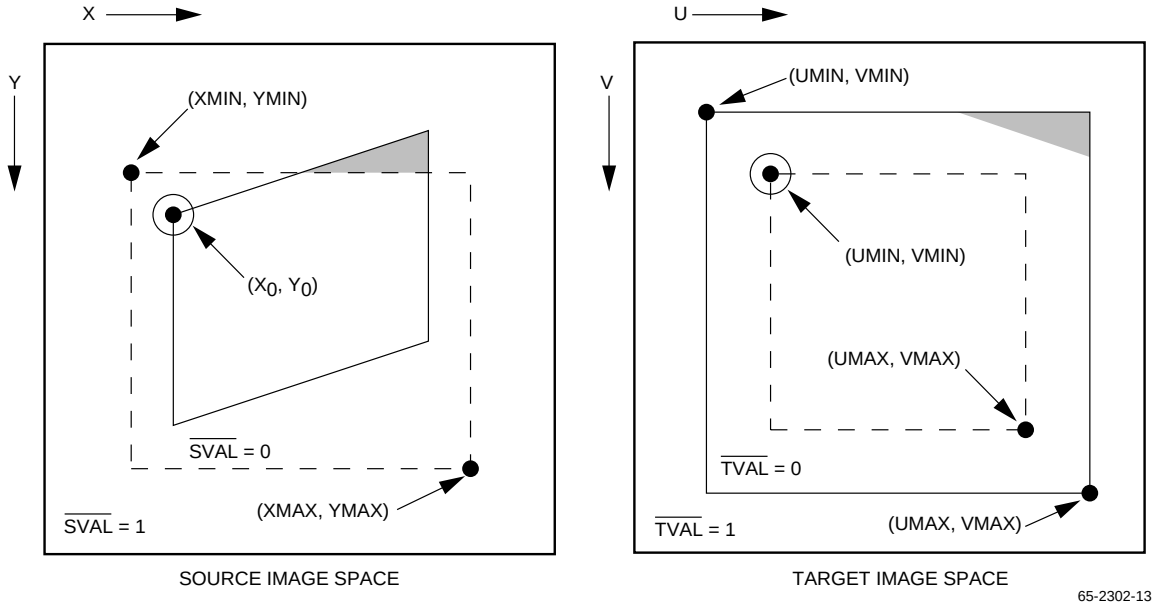
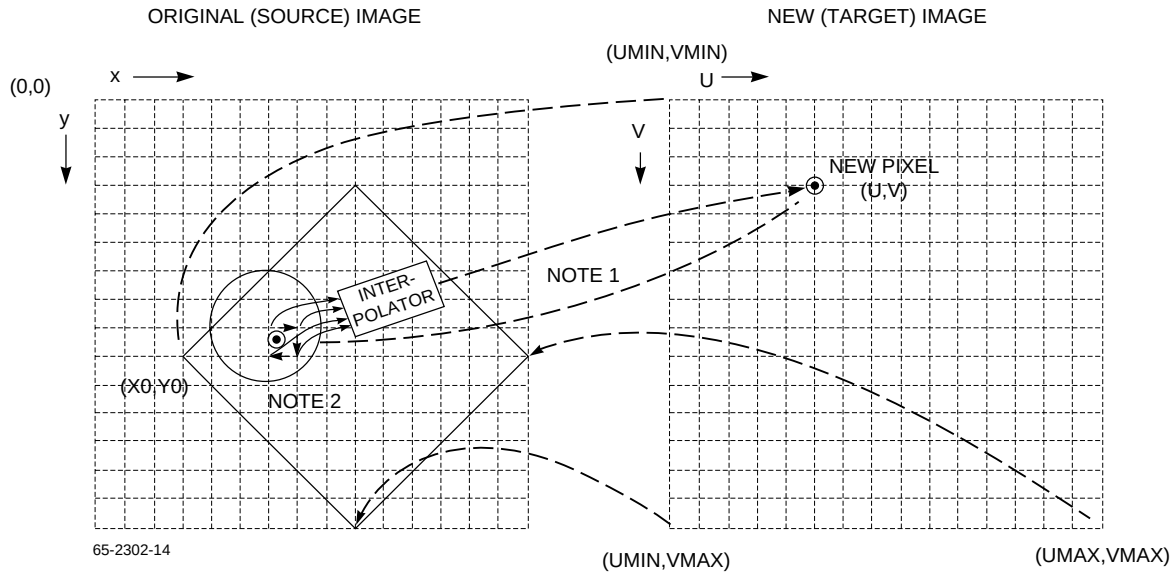


Figure 8. Pixel maps demonstrating source and destination image boundaries, violation flags, and image clipping (note shaded areas)

Real-Time Bilinear Interpolation Using the TMC2302A or TMC2301

Image transformations and translations in bit mapped systems are done by taking an original (source) image, performing coordinate remapping and interpolation, then restoring the image into a new (destination) image space. The coordinates are remapped according to a transformation polynomial. The polynomial, evaluated at destination pixel addresses, maps the transformed pixel addresses (U, V) to pixel addresses in the original image addresses (U,V) to pixel addresses in the original image (X, Y), i.e., (X, Y) is a polynomial function of (U, V).



Notes:

1. Coordinate transformation: Each pixel in (U, V) space is mapped to a location in (X,Y) space.
2. Interpolation: Unless the pixel in (U, V) space coincides with one in (X, Y) space, its amplitude must be estimated as a weighted average of those of the surrounding pixels in (X, Y) space. If the interpolation is done serially, throughput suffers in proportion to the size of the interpolation kernel. However, the interpolation can also be performed in parallel to preserve throughput, as discussed here.

The TMC2302A Image Manipulation Sequencer

The TMC2302A is a controller/address generator, around which an image filtering and resampling system can be built. Under limited supervision from an external controller, the TMC2302A will generate the sequence of memory read and write addresses to transform, resample, and/or filter an image. In all cases, it fetches data from one image buffer, governs its convolution with a user-specified kernel of coefficients, and directs the results to another image memory space. With 24-bit source address buses the device can operate from a source frame size of, for example, 64K X 64K pixels with spatial resolution of 1/256th pixel. A simplified block diagram of the TMC2302A is shown in Figure 9. Although the 24 source addresses bits of each TMC2302A can be designed arbitrarily with the source image address bus, assume for the current discussion that bits SADR (19:8) will correspond to the source image address and that SADR (7:4) therefore denote subpixel postponing to 1/16 pixel resolution.

The basic 2-D system, shown in Figure 10, consists of data source and destination memories, coefficient lookup table, multplier-accumulator, TMC2302A parameters to define the transform and starts the operation. It may also control the loading of the source image into RAM and provide the screen refresh, if needed.

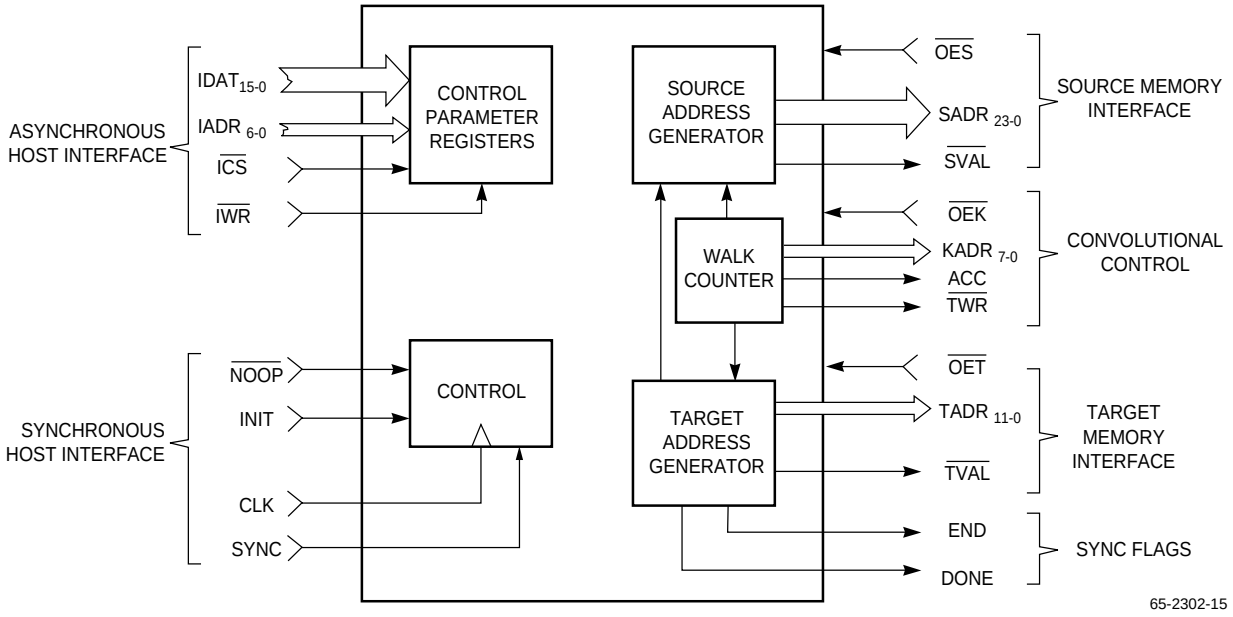


Figure 9. TMC2302A Block Diagram

Preliminary Information

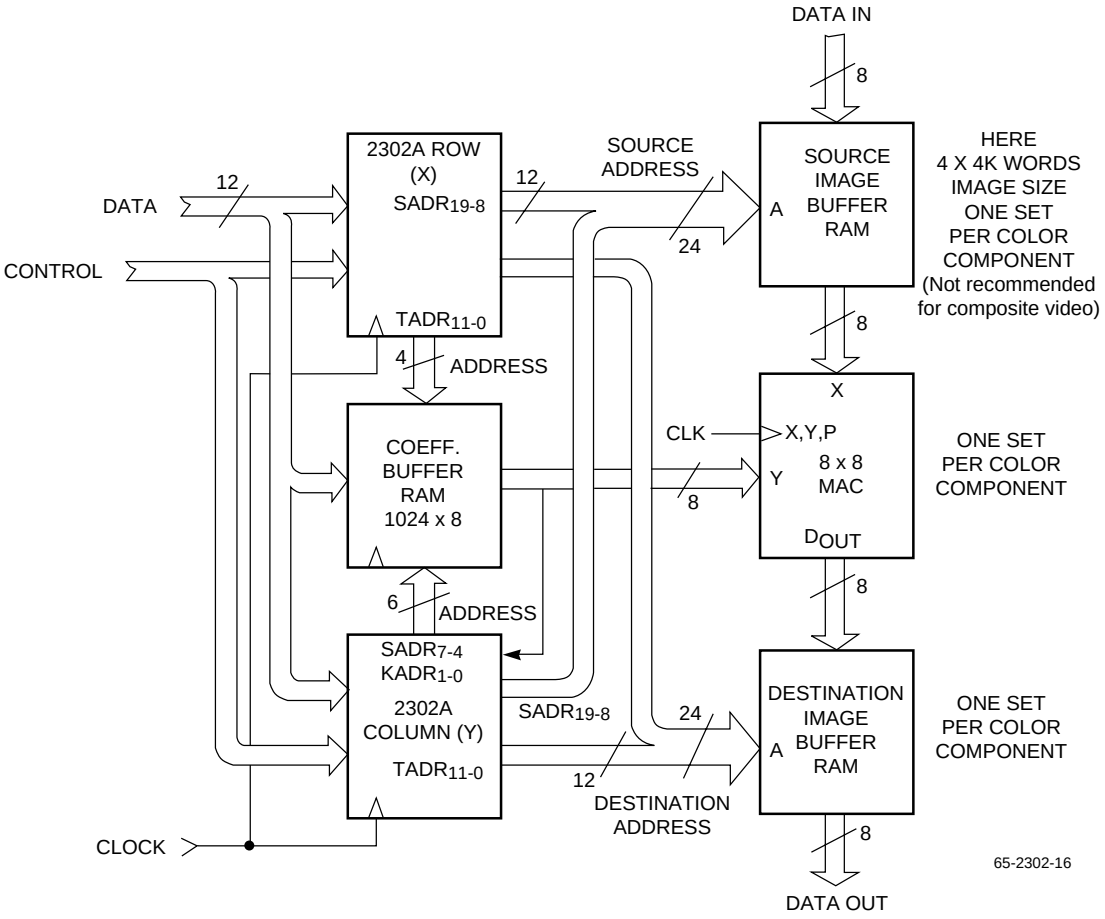


Figure 10. Basic 2-D Image Transformation Systems

Inexact Transformations

In many cases, evaluation of the transformation polynomial results in a non-integer result (non-integer address in the X, Y image space). In such cases, the mapping from original image to transformed image will be inexact. When this occurs, the user has the option of accepting the pixel "nearest" to the address generated, or performing interpolation, a weighted average of nearby pixel values. Using the pixel nearest the address generated is the fastest method since one transformed pixel can be generated on every cycle. The resulting image will include jagged biasing artifacts, however. Performing several transformations on the same image will further degrade the resulting image.

One Cycle Bilinear Interpolation

A better image can be obtained by finding the four pixels nearest the address generated and performing a weighted averaging to determine the new pixel value. This is known as bilinear interpolation. The TMC2302A eases the control logic required for such a function by performing a "walk" around the four closest pixels in the source image space. Essentially, the TMC2302A generates the addresses of the four walk cycles, and the current source pixel is multiplied by a weighting factor and accumulated by the external multiplier accumulator. At the end of the walk, the accumulated result from the four nearest pixels is written into the destination image RAM and the TMC2302A proceeds to the next group. The obvious disadvantage to using bilinear interpolation is that one new destination pixel is generated only on every fourth cycle, reducing the output bandwidth by a factor of four.

One method of "real-time" bilinear interpolation consists of using four memories, each containing the entire source image. The storage arrangement of the pixels within each bank is staggered so that a single address fed to the memories will result in the access of the proper four pixel group. The TMC2302A is programmed to generate the nearest neighbor address and the four nearest pixels are accessed simultaneously and input to the four independent multipliers of a TMC2246 quad multiplier chip. The four pixels are multiplied by their associated weighting factors and added to determine the destination pixel sum. The major drawback of this method is the prohibitive cost for additional memory required to store four copies of the entire source image. For large images, the memory cost and additional board space makes this method unattractive.

A more efficient method is to divide the original source image into a “four-color checker board” and to store it into four separate pixel memory banks, each containing 14th of the source image. Since the image is separated into four memories rather than duplicated, no additional image memory is required. The goal is to separate the image so that any square of four adjacent pixel locations can be accesssed simultaneously. Thus, the user must organize the memory such that the four pixels of any cluster will reside in separate memory banks. With this method, only one set of address generators (TMC2302As) is necessary, and only a slight address modification is necessary to guarantee that the correct group of pixels is accessed and output to the multipliers. Since all pixels are accessed simultaneously, no “walk” is performed, and the TMC2302A system is able to generate one destination pixel on each clock cycle. For example, a 1024 x 768 image can be generated every 20ms for a frame refresh rate of 50Hz. This method which will be described below.

Using Banded Pixel Memory

The TMC2302A should be programmed to do “nearest-neighbor” transformations (Kernel, K = 0 and the X_0 and Y_0 start boundaries programmed without 1/2-LSB truncation debiasing to force address truncation when evaluating the transformation polynomial for the nearest-pixel address). The biased X_0 and Y_0 guarantee that when the exact pixel address falls within the region of four pixels, the upper left-most pixel will always be selected as “nearest-neighbor.”

The key to performing real-time bilinear interpolation is to arrange the pixels in memory so that the four pixels of every grouping will be stored in separate memories. The four nearest pixels will form a square. Figure 11 shows a sample 512 x 512 pixel image and the arrangement into four separate memory banks designated A, B, C, and D. It can be seen from the figure that any (square) grouping of four pixels will have one pixel located in each bank. Thus, one memory sector will hold even row-even column pixels, another, even-row-add column pixels, etc.

$A_{0,0}$ (0,0)	$B_{0,0}$ (1,0)	$A_{1,0}$ (2,0)	$B_{1,0}$ (3,0)	$A_{2,0}$ (4,0)	$B_{2,0}$ (5,0)	$A_{3,0}…A_{255,0}$ (6,0)…(510, 0)	$B_{255.0}$ (511, 0)
$C_{0,0}$ (0,1)	$D_{0,0}$ (1,1)	$C_{1,0}$ (2,1)	$D_{1,0}$ (3,1)	$C_{2,0}$ (4,1)	$D_{2,0}$ (5,1)	$C_{3,0}…C_{255,0}$ (6,1)…(510,1)	$D_{255,0}$ (511,1)
$A_{0,1}$ (0,2)	$B_{0,1}$ (1,1)	$A_{1,1}$ (2,1)	$B_{1,1}$ (3,2)	$A_{2,1}$ (4,2)	$B_{2,1}$ (5,2)	$A_{3,1}…A_{255,1}$ (6,2)…(510,2)	$B_{255,1}$ (511,2)
$C_{0,1}$ (0,3)	$D_{0,1}$ (1,3)	$C_{1,1}$ (2,3)	$D_{1,1}$ (3,3)	$C_{2,1}$ (4,3)	$D_{2,1}$ (5,3)	$C_{3,1}…C_{255,1}$ (6,3)…(510,3)	$D_{255,1}$ (511,3)
$A_{0,2}$ (0,4)	$B_{0,2}$ (1,4)	$A_{1,2}$ (2,4)	$B_{l,2}$ (3,4)	$A_{2,2}$ (4,4)	$B_{2,2}$ (5,4)	$A_{3,2}…A_{255,2}$ (6,4)…(510,4)	$B_{255,2}$ (511,4)
$C_{0,2}$ (0,5)	$D_{0,2}$ (1,5)	$C_{1,2}$ (2,5)	$D_{1,2}$ (3,5)	$C_{2,2}$ (4,5)	$D_{2,2}$ (5,5)	$C_{3,2}…C_{255,2}$ (6,5)…(510,5)	$D_{255,2}$ (511,5)
…	…	…	…	…	…	…	…
$A_{0,255}$ (0,510)	$B_{0,255}$ (1,510)	$A_{1,255}$ (2,510)	$B_{1,255}$ (3,510)	$A_{2,255}$ (4,510)	$B_{2,255}$ (5,510)	$A_{3,255}…A_{255,255}$ (6,510)…(510,510)	$B_{255,255}$ (511,510)
$C_{0,255}$ (0,511)	$D_{0,255}$ (1,511)	$C_{1,255}$ (2,511)	$D_{1,255}$ (3,511)	$C_{2,255}$ (4,511)	$D_{2,255}$ (5,511)	$C_{3,255}…C_{255,255}$ (6,511)…(510,511)	$D_{255,255}$ (511,511)

Figure 11. Source Image Pixel Arrangement

Subscripts i, j for A, B, C, and D denote relative addresses in memory respectively.

The ordered pairs (a, b) denote the physical (X,Y) pixel locations and the TMC2302A SAPR(X) and SADR(Y) address outputs.

The pixels of the original image should be stored in the source RAM banks as shown in Figure 12. The original source image can be loaded by decoding the TMC2302A least significant address bits ($SADR_X(8)$. $SADR_Y(8)$ to determine the memory bank for the pixel while the most-significant address bits ($SADR_X(19:9)$, $SADR_Y(19:9)$) are used as common address lines to all four memory banks.

TMC2302A Address		Bank A	Bank B	Bank C	Bank D
XAμ,	YAμ	$XA_0YA_0 = 00$	$XA_0YA_0 = 10$	$XA_0YA_0 = 01$	$XA_0YA_0 = 11$
0	0	$A_{0,0}$	$B_{0,0}$	$C_{0,0}$	$D_{0,0}$
0	1	$A_{0,1}$	$B_{0,1}$	$C_{0,1}$	$D_{0,1}$
0	2	$A_{0,2}$	$B_{0,2}$	$C_{0,2}$	$D_{0,2}$
0	255	$A_{0,255}$	$B_{0,255}$	$C_{0,255}$	$D_{0,255}$
1	0	$A_{1,0}$	$B_{1,0}$	$C_{1,0}$	$D_{l,0}$
1	1	$A_{1,1}$	$B_{1,1}$	$C_{1,1}$	$D_{1,1}$
1	2	$A_{1,2}$	$B_{1,2}$	$C_{1,2}$	$D_{1,2}$
255	254	$A_{255,254}$	$B_{255,254}$	$C_{255,254}$	$D_{255,254}$
255	255	$A_{255,254}$	$B_{255,254}$	$C_{255,254}$	$D_{255,254}$

In the following discussion, the TMC2302A address outputs $SADR_X$ and $SADR_Y$ will be designated as:

Horizontal Source

XA_0	Least-Significant Horizontal Source X-Address Bit $SADR_X$ (8)
XAμ	Upper Horizontal Source Address Bits $SADR_X$ (19:9)
YA_0	Least-Significant Vertical Source Y-Address Bit $SADR_X$(8)
Yaμ	Upper Vertical-Source Address Bits $SADR_Y$ (19:9)

Interpolation Kernel

$P_{i,j}$ $P_{i,j+1}$

* - actual Pixel

$P_{i+1,j}$ $P_{i+1,j+1}$

Figure 13. TMC2302A Serial Walk Sequence in real time bilinear resampling, this is executed in parallel

When the transformation polynomial is evaluated and the resulting pixel address falls within a group of four nearby pixels (non-integer result), the TMC2302A will always choose the upper leftmost pixel (P_{ij}) as the nearest neighbor (due to the fractional address truncation in the X and Y directions). Since the four pixels will reside in independent banks, the upper leftmost pixel might be located in any of the four memory banks (A,B,C, or D). The bank which contains the nearest neighbor must be known, since in each case, different memory address modification is required to select the correct pixel from each bank.

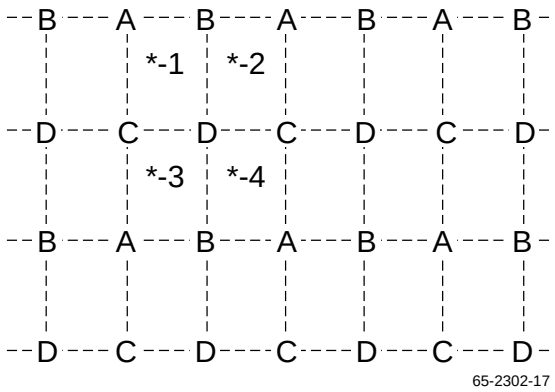


Figure 14. Possible Selections for Nearest Neighbor

Memory Address Modification

Using the address LSBs (XA_0, YA_0) from each TMC2302A external logic can determine which bank contains the nearest neighbor. (This same decoding is used when loading the original image into the source image RAMs.)

Case*	XA_0	YA_0	Nearest Neighbor (Upper Leftmost) Pixel
1	0	0	A Memory Bank contains Nearest Neighbor
2	1	0	B Memory Bank contains Nearest Neighbor
3	0	1	C Memory Bank contains Nearest Neighbor
4	1	1	D Memory Bank contains Nearest Neighbor

*from Figure 14 above

Addressing for each memory bank (A, B, C, D) is done using the uppermost address bits (XA_μ) of the TMC2302As. The LSB of each TMC2302A is used to determine both the upper leftmost pixel and the address modification required. In the following paragraphs, the lower case subscripts (i,j) denote the address of a pixel within a given memory bank (A, B, C, or D), and XA, YA are used to denote physical address outputs of the TMC2302A pairs.

Pixel address modification use to access the correct four pixel group is determined as follows:

Case A:
$A_{i,j}$ is nearest upperleft neighbor,
(No address modifications)
($XA_0 = YA_0 = 0$)

$A_{i,j}$ $B_{i,j}$
*
$C_{i,j}$ $D_{i,j}$

Figure 15. Pixel Memory Mapping for A = Upper Leftmost

Memory Addressing Becomes:
A address = XA_μ, YA_μ
B address = XA_μ, YA_μ
C address = XA_μ, YA_μ
D address = XA_μ, YA_μ
i.e., no modification is required.

Case B:
$B_{i,j}$ is upperleft neighbor,
(Modify X component of A & C memory addresses)
($XA_0 = 1, YA_0 = 0$)

$B_{i,j}$ $A_{i+1,j}$
*
$D_{i,j}$ $C_{i+1,j}$

Figure 16. Pixel Memory Pattern for B = Upper Leftmost

Memory Addressing Becomes:
A address = $(XA_\mu + 1, YA_\mu)$
B address = (XA_μ, YA_μ)
C address = $(XA_\mu + 1, Ya_\mu)$
D address = $(XA_\mu\ YA_\mu)$

Case C:
Ci,j is upperleft neighbor,
(Modify Y component of A & B memory addresses)
($XA_0 = 0, YA_0 = 1$)

$C_{i,J}$ $D_{i,j}$
*
$A_{i,j+1}$ $B_{i,j+1}$

Figure 17. Pixel Pattern for C = Upper Leftmost

Memory Addressing Becomes:
A address = $XA_\mu\ YA_\mu + 1$
B address = $XA_\mu, Ya_\mu + 1$
C address = XA_μ, YA_μ
D address = XA_μ, YA_μ

Case D:
Di,j is the nearest neighbor
(Modify A, B & C addresses, X and Y components)
($XA_0 = 1, YA_0 = 1$)

$D_{i,j}$ $C_{i+1,j}$
*
$B_{i,j+1}$ $A_{i+1,j+1}$

Figure 18. Pixel Pattern for D = Upper Leftmost

Memory Addressing Becomes:
A address = $XA_\mu + 1, YA_\mu + 1$
B address = $XA_\mu, Ya_\mu + 1$
C address = $XA_\mu + 1, YA_\mu$
D address = XA_μ, YA_μ

Taking a close look at the address modifications required for each case above, a simple pattern can be seen. This pattern leads to a set of address modification “rules” based on the values of the least-signficant address bits from the TMC2301s (XA_0 and YA_0). These rules are:

When $YA_0 = 0$. (Case A & B)
No modificaton to the Y address component (YA_μ) is necessary.

When $YA_0 = 1$, (Case C & D)
The Y component (YA_μ) of addresses to the A & B memory banks must be incremented by 1.

When $XA_0 = 0$. (Case A & C)
No modification to the X address component (XA_μ) is necessary.

When $XA_0 = 1$, (Case B & D)
The X component (XA_μ,) of addresses to the A & C memory banks must be incremented by 1.

A system can easily be designed to modify the pixel memory addresses according to the above criteria, to select the correct four pixels to be interpolated. Rather than actually performing a “conditional” address increment as discussed above. It requires less logic simply to add the LSB address bit to the memory bank addresses (XA_μ, YA_μ). Figure 12 shows the logic to perform the required address modifications. The addition (XA_μ, + XA_0, YA_μ, +YA_0) can be done using half-adders with the XA_0 (YA_0) address output of the TMC2302A connected to the carry-in of each adder. It can also be done using high-speed programmable logic.

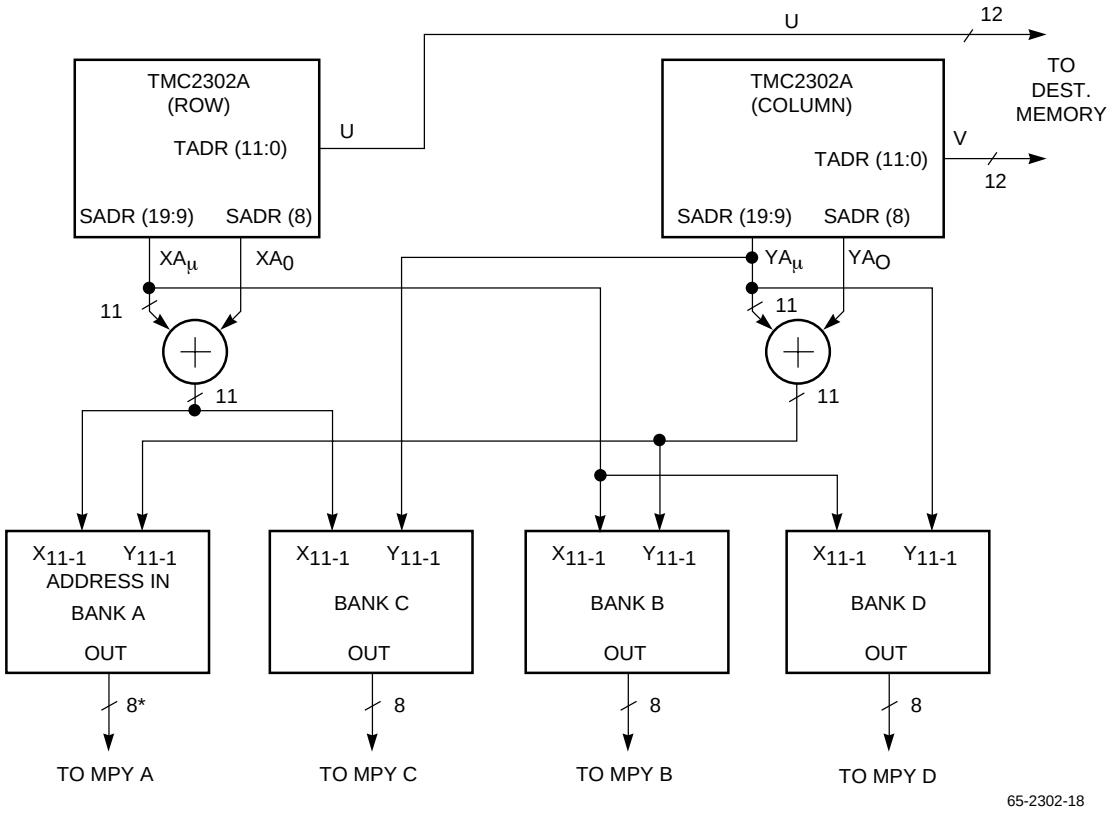
Note: Only modifications to the source image memory are necessary. The destination image memory may be arranged in a linear or other type array as required by the refresh circuitry.

Coefficient Memory

Typically, the 4 highest fractional source address bits from each TMC2302A (SADR (7:4) in the example) are used to reflect the offset from the nearest XA (YA) pixel location.

With spatial resolution of 4 bits in both the X and Y directions, there can be as many as 256 unique coefficient values. This requires a coefficient memory of at least 256 bytes. However, as shown below, each of the four different cases requires its own set of 256 coefficients.

One-cycle bilinear interpolation requires four independent coeffficient memories, so that a parallel multiplication can be performed with the four pixel values.



*Number of bits of intensity per pixel, per column component, typically 8 to 12.

Figure 19. Memory Address Modification

Preliminary Information

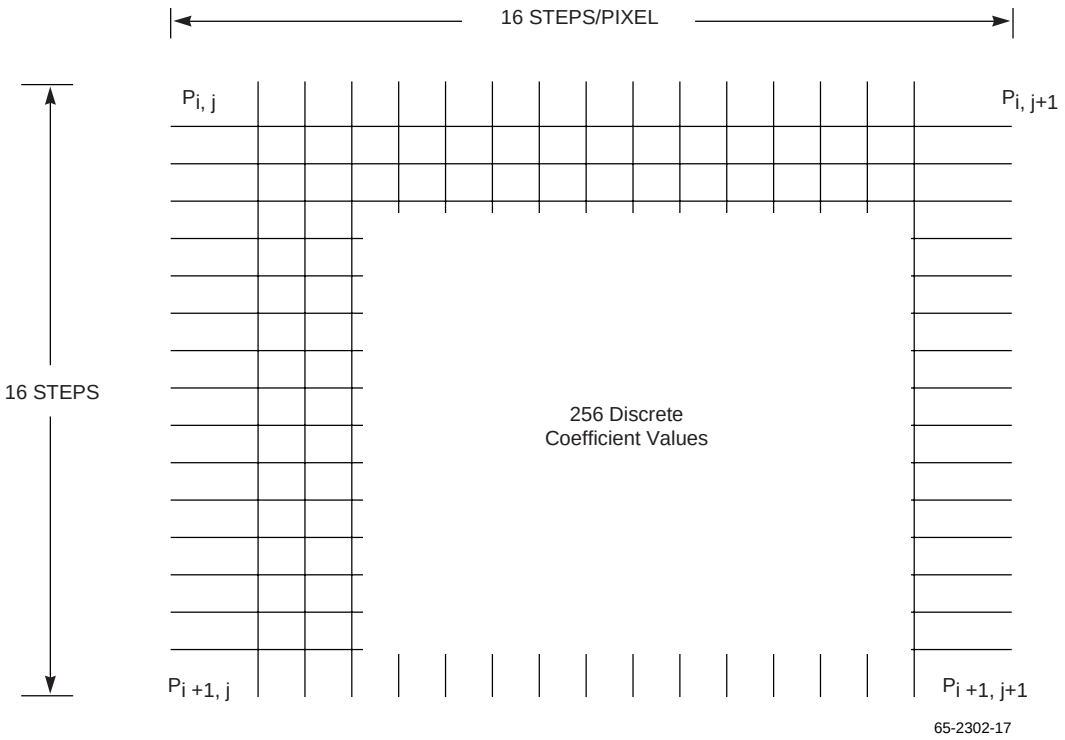


Figure 20. Intrapixel Resolution

Similar to determining the correct four pixel group, the coefficients must take into account the memory bank (A, B, C, or D) that contains the upper leftmost pixel, and adjust the coefficients accordingly. These adjustments are necessary since the fractional address outputs ($SADR_X$ 7:4), $SADR_Y$ (7,4) from the TMC2302As reflect the spatial distance only from the upper leftmost pixel within the pixel group. Assuming that the fractional addresses $SADR_X$ (7:4) and $SADR_Y$ (7:4) plus the integer LSBs $SADR_X$ (8) and $SADR_Y$ (8) are to be used directly to address the 1024-byte coefficient memory, the loading of the coefficients is shown below with F_X = $SADR_X$ (7:4) and F_Y = $SADR_Y$ (7:4) Case A through D are the same as discussed previously for the pixel address modifications.

Case A:
A is nearest neighbor ($XA_0 = 0, YA_0 = 0$)

Coeff A = $(1 - f_X) * (1 - f_Y)$
Coeff B = $(f_X) *(1 - f_Y)$
Coeff C = $(1 - f_X) * (f_Y)$
Coeff D = $f_X * f_Y$

Case B:
B is nearest neighbor ($XA_0 = 1, YA_0 = 0$)

Coeff A = $f_X * (1-f_Y)$
Coeff B =$(1-f_X) * (1-f_Y)$
Coeff C = $f_X * f_Y$
Coeff D = $(l-f_X)f_Y$

Case C:
C is nearest neighbor ($XA_0 = 0, YA_0 = 1$)

Coeff A = $(1- f_X) f_Y$
Coeff B = $f_X f_Y$
Coeff C = $(1 - f_X) (1 - f_Y)$Coeff D = $f_X * (1 - f_Y)$

Case D:
D is nearest neighbor ($XA_0 = 1, YA_0 = 1$)

Coeff A = $f_X f_Y$
Coeff B = $(1 - f_X)f_Y$
Coeff C = $f_X * (1 - f_Y)$
Coeff D = $(1 - f_X) (1 - f_Y)$

Incorporating the concepts outlined in this discussion, the final system for one-cycle blinear interpolation is shown in Figure 21. This figure shows a small increase in logic over the basic 2-D system shown in Figure 10. The additional logic required includes: TMC2246 (rather than a single multiply/accumulate), and three additional coefficient memories. Some additional decoding logic is required to load the four pixel memory banks as well as some data and address pipelining (registering) to meet timing requirements. The solution, however, provides an increased pixel bandwidth, by a factor of four, and only a small increase in part count.

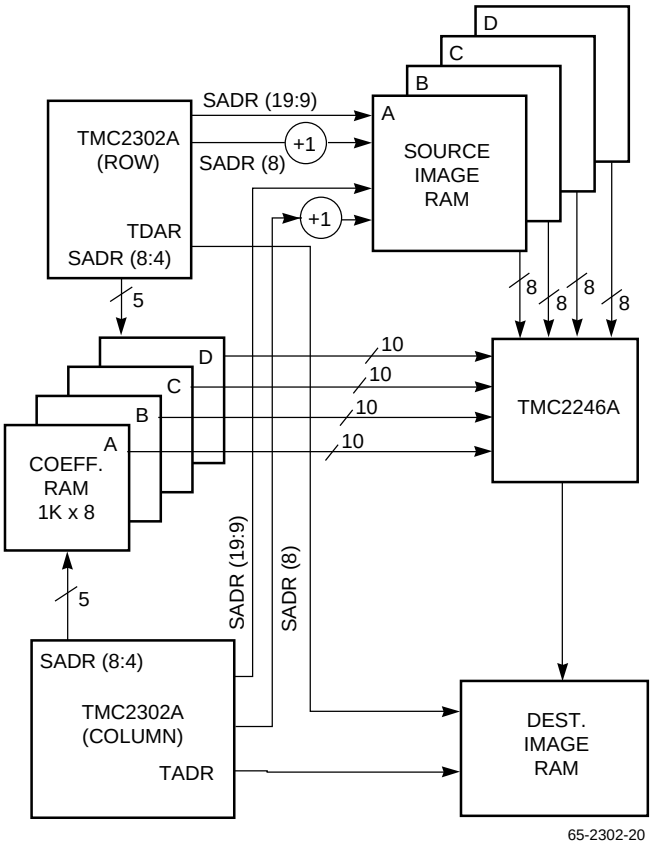


Figure 14. One-Cycle Bilinear Interpolation System

Related Products

- TMC2301 Image Resampling Sequencer
- TMC2246A Image Filter
- TMC2249A Digital Mixer
- TMC2242B Half-Band Filter

Notes:

Notes:

Notes:

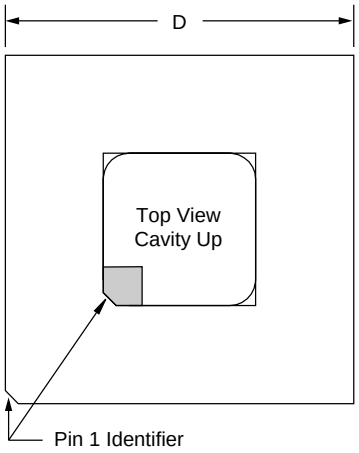
Mechanical Dimensions

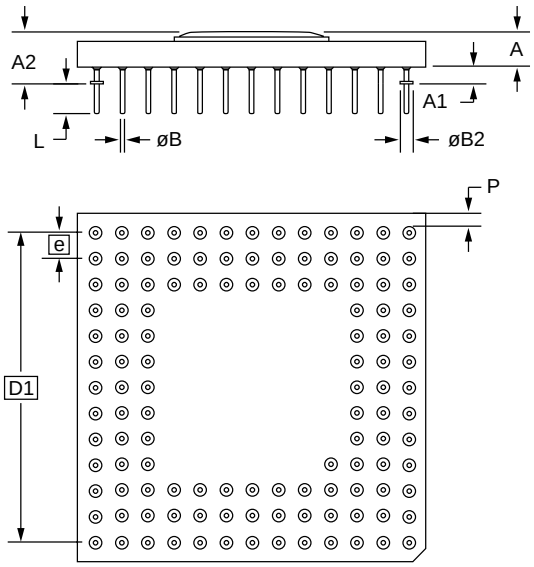
121-Lead PPGA Package

Symbol	Inches		Millimeters		Notes
	Min.	Max.	Min.	Max.	
A	.080	.160	2.03	4.06	
A1	.040	.060	1.01	1.53	
A2	.125	.215	3.17	5.46	
øB	.016	.020	0.40	0.51	2
øB2	.050 NOM.		1.27 NOM.		2
D	1.340	1.380	33.27	35.05	SQ
D1	1.200 BSC		30.48 BSC		
e	.100 BSC		2.54 BSC		
L	.110	.145	2.79	3.68	
L1	.170	.190	4.31	4.83	
M	13		13		3
N	120		120		4
P	.003	—	.076	—	

Notes:

1. Pin #1 identifier shall be within shaded area shown.
2. Pin diameter excludes solder dip finish.
3. Dimension "M" defines matrix size.
4. Dimension "N" defines the maximum possible number of pins.
5. Orientation pin is at supplier's option.
6. Controlling dimension: inch.





Preliminary Information

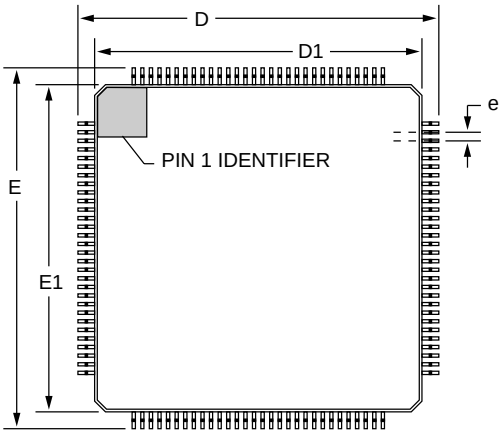
Mechanical Dimensions (continued)

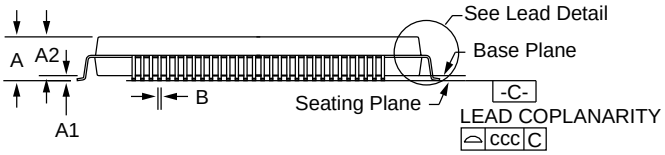
120-Lead MQFP Package

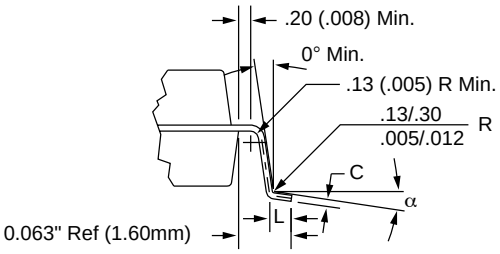
Symbol	Inches		Millimeters		Notes
	Min.	Max.	Min.	Max.	
A	—	.154	—	3.92	
A1	.010	—	.25	—	
A2	.125	.144	3.17	3.67	
B	.012	.018	.30	.45	3, 5
C	.005	.009	.13	.23	5
D/E	1.219	1.238	30.95	31.45	
D1/E1	1.098	1.106	27.90	28.10	
e	.0315 BSC		.80 BSC		
L	.026	.037	.65	.95	4
N	120		120		
ND	30		30		
α	0°	7°	0°	7°	
ccc	—	.004	—	.10	

Notes:

1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Controlling dimension is millimeters.
3. Dimension "B" does not include dambar protrusion. Allowable dambar protrusion shall be .08mm (.003in.) maximum in excess of the "B" dimension. Dambar cannot be located on the lower radius or the foot.
4. "L" is the length of terminal for soldering to a substrate.
5. "B" & "C" includes lead finish thickness.







Lead Detail

Preliminary Information

Ordering Information

Product Number	Temperature Range	Speed Grade	Screening	Package	Package Marking
TMC2302AH5C	0°C to 70°C	30 MHz	Commercial	120 Pin Plastic Pin Grid Array	2302AH5C
TMC2302AH5C1	0°C to 70°C	40 MHz	Commercial	120 Pin Plastic Pin Grid Array	2302AH5C1
TMC2302AKEC	0°C to 70°C	30 MHz	Commercial	120 Lead Metric Quad FlatPack	2302AKEC
TMC2302AKEC1	0°C to 70°C	40 MHz	Commercial	120 Lead Metric Quad FlatPack	2302AKEC1

Preliminary Information