Data Sheet May 1999 File Number 2456.5

## Two Dimensional Convolver

The Intersil HSP48908 is a high speed Two Dimensional Convolver which provides a single chip implementation of a video data rate 3 x 3 kernel convolution on two dimensional data. It eliminates the need for external data storage through the use of the on-chip row buffers which are programmable for row lengths up to 1024 pixels.

There are Internal Register banks for storing two independent 3 x 3 filter kernels, thus facilitating the implementation of adaptive filters and multiple filter operations on the same data. The pixel data path also includes an on-chip ALU for performing real-time arithmetic and logical pixel point operations.

Data is provided to the HSP48908 in a raster scan noninterlaced fashion, and is internally buffered on images up to 1024 pixels wide for the 3 x 3 convolution operation. Images with larger rows and convolution with larger kernel sizes can be accommodated by using external row buffers and/or multiple HSP48908s. Coefficient and pixel input data are 8-bit signed or unsigned integers, and the 20-bit convolver output guarantees no overflow for kernel sizes up to 4 x 4. Larger kernel sizes can be implemented however, since the filter coefficients will normally be less than their maximum 8-bit values.

The HSP48908 is manufactured using an advanced CMOS process, and is a low power fully static design. The configuration of the device is controlled through a standard microprocessor interface and all inputs/outputs are TTL compatible.

## Features

- Single Chip 3 x 3 Kernel Convolution
- Programmable On-Chip Row Buffers
- DC to 32MHz Clock Rate
- Cascadable for Larger Kernels and Images
- On-Chip 8-Bit ALU
- Dual Coefficient Mask Registers, Switchable in a Single Clock Cycle
- 8-Bit Signed or Unsigned Input and Coefficient Data
- 20-Bit Extended Precision Output
- Standard µP Interface
- Low Power CMOS

## Applications

- Image Filtering
- Edge Detection
- Adaptive Filtering
- Real Time Video Filter

## Ordering Information

| PART NUMBER | TEMP. RANGE ($^{o}$C) | PACKAGE | PKG. NO. |
|---|---|---|---|
| HSP48908VC-20 | 0 to 70 | 100 Ld MQFP | Q100x14x20 |
| HSP48908VC-32 | 0 to 70 | 100 Ld MQFP | Q100x14x20 |
| HSP48908JC-20 | 0 to 70 | 84 Ld PLCC | N84.1.15 |
| HSP48908JC-32 | 0 to 70 | 84 Ld PLCC | N84.1.15 |
| HSP48908GC-20 | 0 to 70 | 84 Ld PGA | G84.A |
| HSP48908GC-32 | 0 to 0 | 84 Ld PGA | G84.A |

CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.

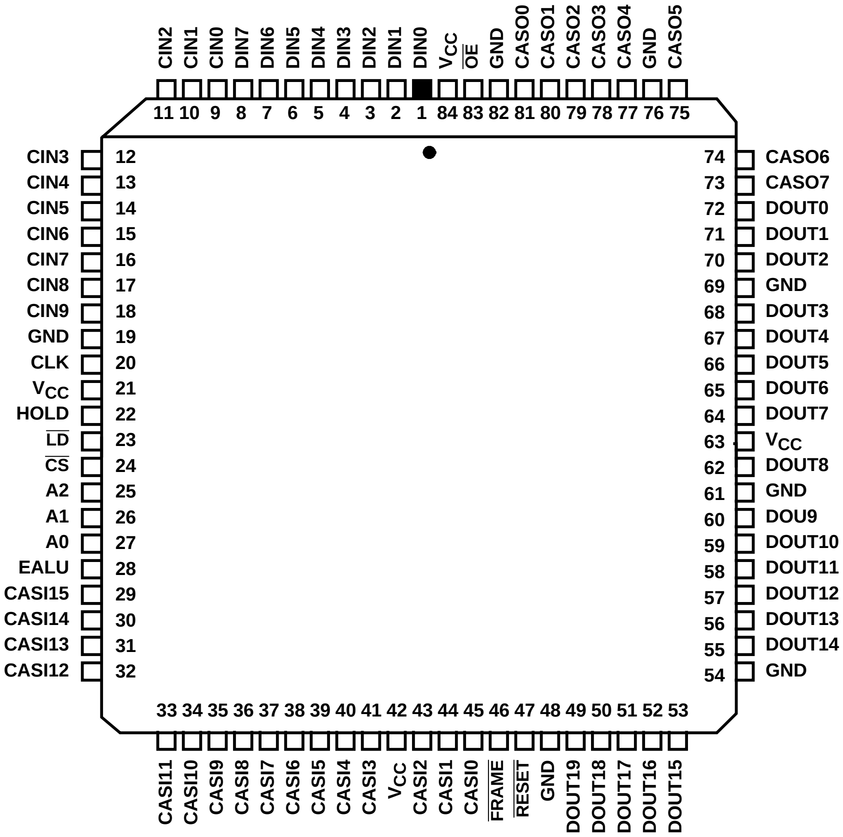
## Pinouts

**84 PIN PGA**
TOP VIEW

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 11 | CAS06 | DOUT0 | DOUT1 | GND | DOUT5 | DOUT6 | DOUT8 | DOUT10 | DOUT12 | DOUT13 | DOUT15 |
| 10 | CAS04 | CAS06 | CAS07 | DOUT2 | DOUT4 | DOUT9 | GND | DOUT11 | DOUT14 | GND | DOUT17 |
| 9 | CAS03 | GND | | | DOUT3 | DOUT7 | $V_{CC}$ | | | DOUT16 | DOUT18 |
| 8 | CAS01 | CAS02 | | | | | | | | DOUT19 | GND |
| 7 | $\overline{OE}$ | GND | $V_{CC}$ | | | | | | CASI1 | $\overline{FRAME}$ | CASI0 |
| 6 | DIN1 | CASO0 | DIN0 | | | | | | CASI2 | $V_{CC}$ | $\overline{RESET}$ |
| 5 | DIN2 | DIN3 | DIN4 | | | | | | CASI6 | CASI14 | CASI13 |
| 4 | DIN5 | DIN6 | | | | | | | | CASI7 | CASI16 |
| 3 | DIN7 | CIN1 | | | CIN9 | HOLD | $\overline{L0}$ | | | CASI10 | CASI18 |
| 2 | CIN0 | CIN3 | CIN4 | CIN7 | GND | $V_{CC}$ | A2 | EALU | CASI13 | CASI11 | CASI9 |
| 1 | CIN2 | CIN5 | CIN6 | CIN8 | CLK | A1 | $\overline{CS}$ | A0 | CASI16 | CASI14 | CASI12 |
| | A | B | C | D | E | F | G | H | J | K | L |

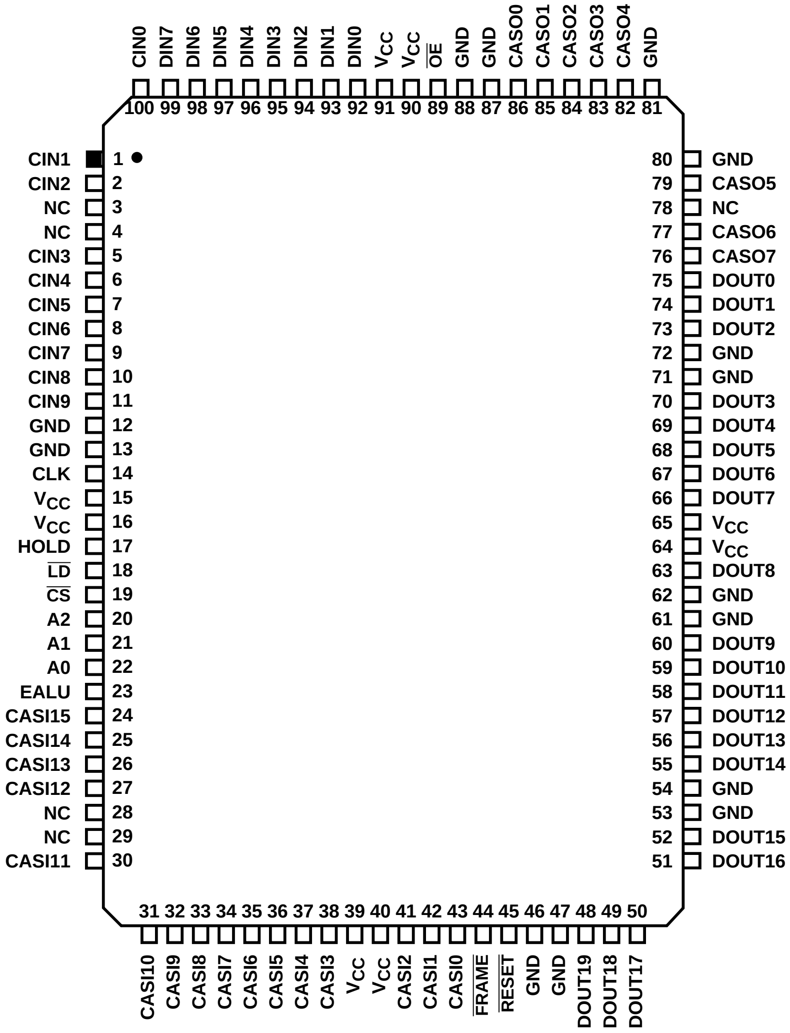
## Pinouts (Continued)

**84 LEAD PLCC**
TOP VIEW

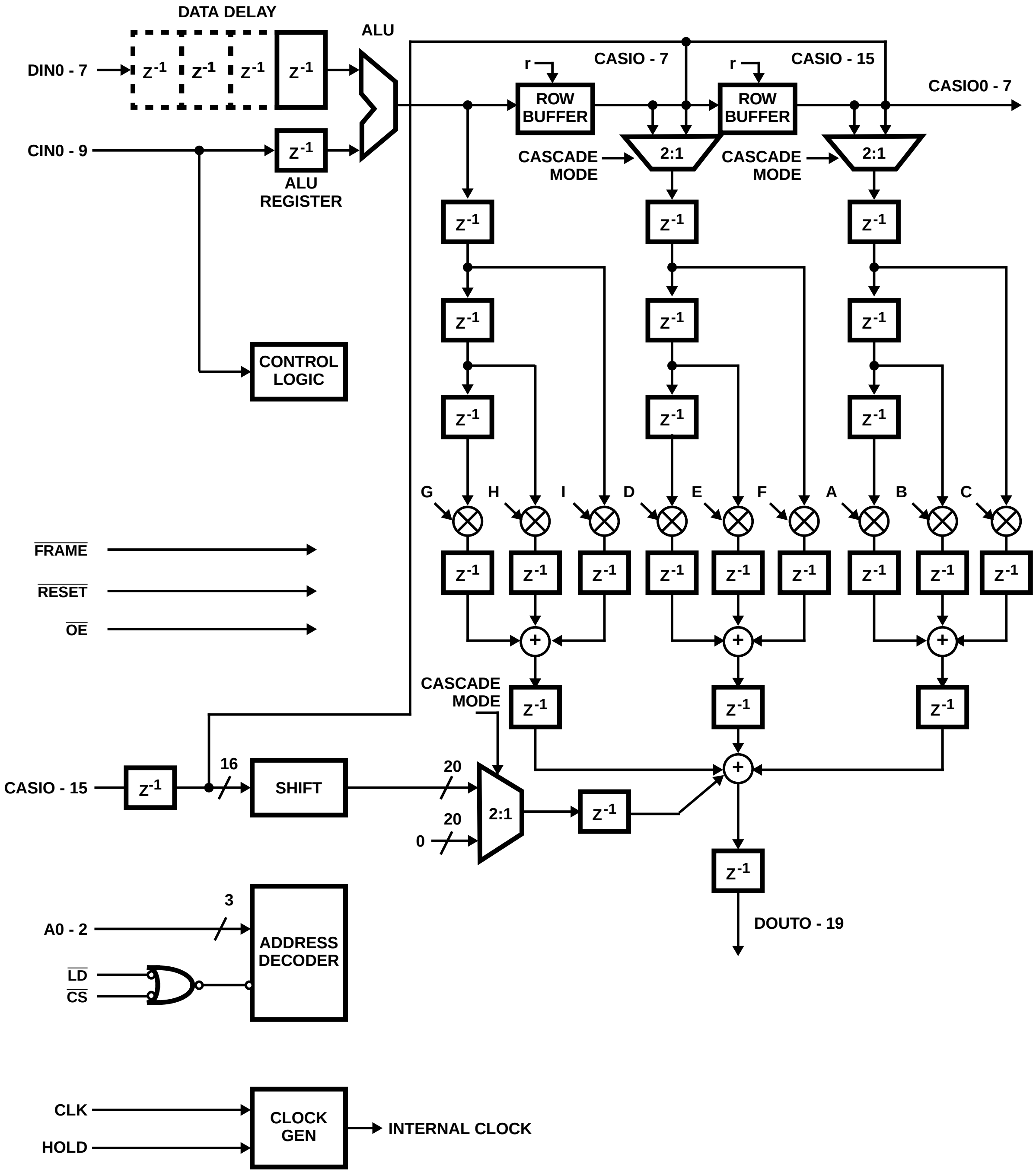


## Pinouts (Continued)

**100 LEAD MQFP**
TOP VIEW



## Block Diagram



## Pin Descriptions

| NAME | PLCC PIN | TYPE | DESCRIPTION |
|---|---|---|---|
| $V_{CC}$ | 21, 42, 63, 84 | | The +5V power supply pins. 0.1µF capacitors between the $V_{CC}$ and GND pins are recommended. |
| GND | 19, 48, 54, 61,<br>69, 76, 82 | | The device ground. |
| CLK | 20 | I | Input and System Clock. Operations are synchronous with the rising edge of this clock signal. |
| DIN-07 | 1-8 | I | Pixel Data Input Bus. This bus is used to provide the 8-bit pixel input data to the HSP48908. The data must be provided in a synchronous fashion, and is latched on the rising edge of the CLK signal. |
| CIN0-9 | 9-18 | I | Coefficient Input Bus. This input bus is used to load the Coefficient Mask Register(s), the Initialization Register, the Row Buffer Length Register and the ALU microcode. It may also be used to provide a second operand input to the ALU. The definition of the CIN0-9 bits is defined by the register address bits A0-2. The CIN0-9 data is loaded to the Addressed Register through the use of the $\overline{\text{CS}}$ and $\overline{\text{LD}}$ inputs. |
| DOUT0-19 | 49-53, 55-60,<br>62, 64-68,<br>70-72 | 0 | Output Data Bus. This 20-Bit output port is used to provide the convolution result. The result is the sum of products of the input data samples and their corresponding coefficients. The Cascade inputs CASI0-15 may also be added to the result by selecting the appropriate cascade mode in the Initialization Register. |
| CASIO-15 | 29-41, 43-45 | I | Cascade Input Bus. This bus is used for cascading multiple HSP48908s to allow convolution with larger kernels or row sizes. It may also be used to interface to external row buffers. The function of this bus is determined by the Cascade Mode bit (Bit 0) of the Initialization Register. When this bit is set to a ‘0’, the value on CASI0-15 is left shifted and added to DOUT0-19. The amount of the shift is determined by bits 7-8 of the Initialization Register. While this mode is intended primarily for cascading, it may also be used to add an offset value, such as to increase the brightness of the convolved image.<br>When the Cascade mode bit is set to a “1”**,** this bus is used for interfacing to external row buffers. In this mode the bus is divided into two 8-bit busses (CASI0-7 and CASI8-15), thus allowing two additional pixel data inputs. The cascade data is sent directly to the internal multiplier array which allows for larger row sizes without using multiple HSP48908s. |
| CASO0-7 | 73-75, 77-81 | 0 | Cascade Output Bus. This bus is used primarily during cascading to handle larger frames and/or kernel sizes. This output data is the data on DIN0-7 delayed by twice the programmed internal row buffer length. |
| $\overline{\text{FRAME}}$ | 46 | I | $\overline{\text{Frame}}$ is an asynchronous new frame or vertical sync input. A low on this input resets all internal circuitry except for the Coefficient, ALU, AMC, EOR and INT Registers. Thus, after a $\overline{\text{Frame}}$ reset has occurred, a new frame of pixels may be convolved without reloading these registers. |
| EALU | 28 | I | Enable ALU Input. This control line gates the clock to the ALU Register. When it is high, the data on CIN0-7 is loaded on the next rising clock edge. When EALU is low, the last value loaded remains in the ALU Register. |
| HOLD | 22 | I | The Hold Input is used to gate the clock from all of the internal circuitry of the H5P48908. This signal is synchronous, is sampled on the rising edge of CLK and takes effect on the following cycle. While this signal is active (high), the clock will have no effect on the HSP48908 and internal data will remain undisturbed. |
| $\overline{\text{RESET}}$ | 47 | I | Reset is an asynchronous signal which resets all internal circuitry of the HSP48908. All outputs are forced low in the reset state. |
| $\overline{\text{OE}}$ | 83 | I | Output Enable. The $\overline{\text{OE}}$ input controls the state of the Output Data bus (DOUT0-19). A LOW on this control line enables the port for output. When $\overline{\text{OE}}$ is HIGH, the output drivers are in the high impedance state. Processing is not interrupted by this pin. |
| A0-2 | 25-27 | I | Control Register Address. These lines are decoded to determine which register in the control logic is the destination for the data on the CIN0-9 inputs. Register loading is controlled by the A0-2, $\overline{\text{LD}}$ and $\overline{\text{CS}}$ inputs. |
| $\overline{\text{LD}}$ | 23 | I | Load Strobe. $\overline{\text{LD}}$ is used for loading the Internal Registers of the HSP48908. When $\overline{\text{CS}}$ and $\overline{\text{LD}}$ are active, the rising edge of $\overline{\text{LD}}$ will latch the CIN0-7 data into the register specified by A0-2. |
| $\overline{\text{CS}}$ | 24 | I | Chip Select. The Chip Select input enables loading of the Internal Registers. When $\overline{\text{CS}}$ is low, the A0-2 address lines are decoded to determine the meaning of the data on the CIN0-7 bus. The rising edge of $\overline{\text{LD}}$ will then load the Addressed Register. |

## Functional Description

The HSP48908 two-dimensional convolver performs convolution of 3 x 3 filter kernels. It accepts the image data in raster scan, non-interlaced format, convolves it with the filter kernel and outputs the filtered image. The input and filter kernel data are both 8 bits, while the output data is 20 bits to prevent overflow during the convolution operation. The HSP48908 has internal storage for two 3 x 3 filter kernels and is capable of buffering two 1024 x 8-bit rows for true single chip operation at video frame rates. An 8-bit ALU in the input pixel data path allows the user to perform arithmetic and logical operations on the input data in real time during the convolution. Multiple devices can also be cascaded together for larger kernel convolution, larger frame sizes and increased precision.

Image data is input to the convolver via the DIN0-7 bus. The data is then operated on by the ALU, stored in the row buffers and convolved with the 3 x 3 array of filter coefficients. The resultant output data is then latched into the Output Register. The row buffers are preprogrammed to the length of one row of the input image to enable the user to input the image data one pixel at a time in raster scan format without having to provide external storage.

Initialization of the convolver is done using the CIN0-7 bus to load configuration data, such as the filter kernel(s) and the length of the row buffers. The address lines A0-2 are used to address the Internal Registers for initialization. The configuration data is loaded using the A0-2, CIN0-9, $\overline{CS}$ and $\overline{LD}$ controls as address, data, chip select and write enable, respectively. This interface is compatible with standard microprocessors without the use of any additional glue logic.

Filtered image data comes out of the convolver over the DOUT0-1 9 bus. This output bus is 20 bits wide to provide room for growth during the convolution operation. The 20-bit bus will allow the use of up to 4 x 4 kernels (using multiple 48908s) without overflow. However, in practical applications, much larger kernel sizes can be implemented without overflow since the filter coefficients are typically much smaller than 8-bit full scale values. DOUT0-19 is also a registered, three state bus to facilitate cascading multiple chips and to allow the HSP48908 to reside on a standard microprocessor system bus.

Multiple convolvers can also be cascaded together for kernel sizes larger than 3 x 3 and for convolution on images with row lengths longer than 1024 pixels. The maximum kernel size is dependent upon the magnitude of the image data and the coefficients in a given application; care must always be taken with very large kernel sizes to prevent overflow of the 20-bit output.

### Data Input

Image data coming into the 2D Convolver passes through a programmable pipeline delay before being sent to the ALU. The amount of delay (1 to 4 clock cycles) is set in the Initialization Register during configuration setup (See Control Logic). Delays greater than one are used primarily in cascading multiple HSP48908s to align data sequences for proper output (See Operation).

### Arithmetic Logic Unit

The on-chip ALU provides the user with the capability of performing pixel point operations on incoming image data. Depending on the instruction in the ALU Microcode Register, the ALU can perform any one of 19 arithmetic and logical functions, and shift the resulting number left or right by up to 3 bits. Tables 1 and 2 show the available ALU functions and the 10-bit associated microcode to be loaded into the ALU Microcode Register. Note that the shifts take place on the output of the ALU and are completely independent of the logical or arithmetic operation being performed. The first input (A) of the ALU is taken from the pixel input bus (DIN0-7). The second input (B) is taken from the ALU Register. The ALU Register is loaded via the CIN0-7 bus while the EALU control line is valid (see EALU).

**TABLE 1. ALU SHIFT OPERATIONS**

| ALU MICROCODE REGISTER | | | |
|---|---|---|---|
| REGISTER BIT | | | |
| 9 | 8 | 7 | OPERATION |
| 0 | 0 | 0 | No Shift (Default) |
| 0 | 0 | 1 | Shift Right 1 |
| 0 | 1 | 0 | Shift Right 2 |
| 0 | 1 | 1 | Shift Right 3 |
| I | 0 | 0 | Shift Left 1 |
| 1 | 0 | 1 | Shift Left 2 |
| 1 | 1 | 0 | Shift Left 3 |
| 1 | 1 | 1 | Not Valid |

**TABLE 2. ALU PIXEL OPERATIONS**

| REGISTER BIT | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6 | 5 | 4 | 3 | 2 | 1 | 0 | OPERATION |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | Logical (00000000) |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | Logical (11111111) |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | Logical (A) (Default) |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | Logical (B) |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | Logical ($\overline{A}$) |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | Logical ($\overline{B}$) |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | Arithmetic (A + B) |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | Arithmetic (A -B) |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | Arithmetic (B -A) |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | Logical (A AND B) |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | Logical (A AND $\overline{B}$) |

TABLE 2. ALU PIXEL OPERATIONS (Continued)

| REGISTER BIT | | | | | | | OPERATION |
|---|---|---|---|---|---|---|---|
| 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | Logical ($\overline{A}$ AND B) |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | Logical (A OR B) |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | Logical (A OR $\overline{B}$) |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | Logical ($\overline{A}$ OR B) |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | Logical (A NAND B) |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | Logical (AN OR B) |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | Logical (A XOR B) |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | Logical (A XNOR B) |

### EALU

The EALU control pin enables loading of the ALU Register. While the EALU line is high, the data on CIN0-7 is latched into the ALU Register on the rising edge of CLK. When EALU goes low, the current value in the ALU Register is held until EALU is again asserted. Note that the ALU loading operation makes use of the CIN0-7 inputs, but is completely independent of $\overline{CS}$ and $\overline{LD}$. Therefore, in order to prevent overwriting an internal register, care must be taken to ensure that $\overline{CS}$ and $\overline{LD}$ are not active during an EALU cycle.

### Programmable Row Buffers

The programmable row buffers are used for buffering raster input data for the convolution operation. They can be thought of as Programmable Shift Registers which can each store up to 1024 8-bit values, thus, delaying each pixel by up to 1024 clock cycles. Functionally, each row buffer can be represented as a set of registers connected as a 1024 x 8-bit Serial Shift Register. The output of each buffer can be represented by the equation Q = D(n-r), where Q is the row buffer output, D is the buffer input, n is the current clock cycle and r is the preprogrammed row length of the input image. Since the two buffers are connected in series, the data at the cascade outputs (CASO0-7) is delayed by two row delays and may be used for cascading multiple convolvers for larger kernel sizes and/or row lengths. The programmable row buffers can also be bypassed by selecting the appropriate cascade mode in the Initialization Register. This mode allows the use of external row buffers for convolving with row lengths longer than 1024 pixels.

### 8-BIt Multiplier Array

The multiplier array consists of nine 8 x 8 multipliers. Each multiplier forms the product of a filter coefficient with a corresponding pixel in the input image. Input and coefficient data may be in either two's complement or unsigned integer format. The nine coefficients form a 3 x 3 filter kernel which is multiplied by the input pixel data and summed to form a sum of products for implementation of the convolution operation as shown below:

| INPUT DATA | | FILTER KERNEL | |
|---|---|---|---|
| P1 | P2 | P3 | A B C |
| P4 | P5 | P6 | D E F |
| P7 | P8 | P9 | G H I |

| | | | |
|---|---|---|---|
| OUTPUT = | (A x P1) | + (B x P2) | + (C x P3) |
| | + (D x P4) | + (E x P5) | + (F x P6) |
| | + (G x P7) | + (H x P8) | + (I x P9) |

### Control Logic

The control logic (Figure 1) contains the ALU Microcode Register, the Initialization Register, the Row Length Register, and the Coefficient Registers. The control logic is updated by placing data on the CIN0-9 bus and using the A0-2, $\overline{CS}$ and $\overline{LD}$ control lines to write to the Addressed Register (see Address Decoder). All of the Control Logic Registers are loaded with their default values on $\overline{RESET}$, and are unaffected by $\overline{FRAME}$.

### ALU Microcode Register

The ALU Microcode Register is used to store the command word for the ALU. The ALU command word is a 10-bit instruction divided into two fields: the lower 7 bits determine the ALU operation and the upper 3 bits specify the number of shifts which occur. The ALU command words are defined in Tables 1 and 2 (See ALU Section).

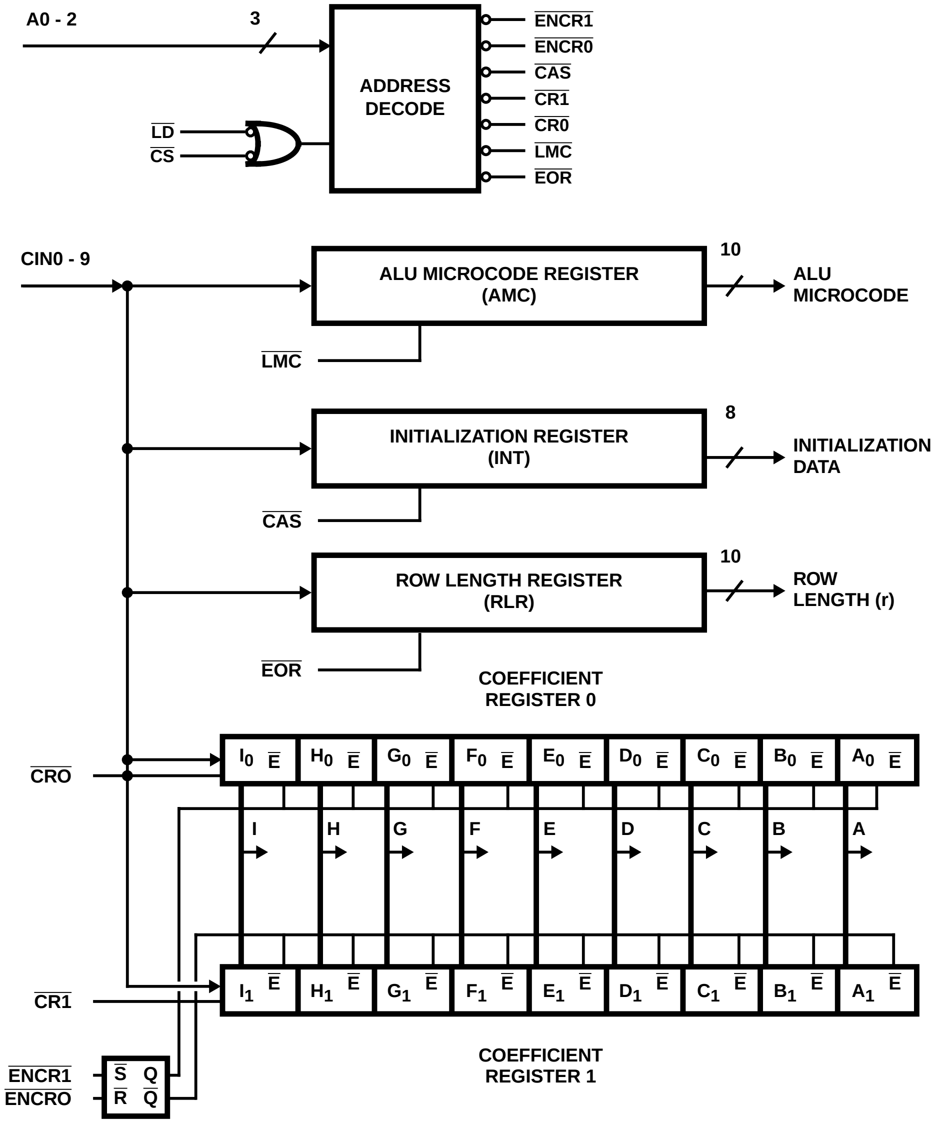


FIGURE 1. CONTROL LOGIC BLOCK DIAGRAM

### Initialization Register

The Initialization Register is used to appropriately configure the convolver for a particular application. It is loaded through the use of the CIN0-7 bus along with the $\overline{CS}$ an $\overline{LD}$ inputs. Bit 0 defines the type of cascade mode to be used; Bits 1 and 2 select the number of delays to be included in the input pixel data path; Bits 3 and 4 define the input and coefficient data format; Bits 5 and 6 determine the type of rounding to occur on the DOUT0-19 bus; Bits 7 an 8 define the shift applied to the cascade input data. The complete definition of the Initialization Register bits is give in Table 3.

**TABLE 3. INITIALIZATION REGISTER DEFINITION**

| INITIALIZATION REGISTER | | |
|---|---|---|
| **BIT 0** | | **FUNCTION = CASCADE MODE** |
| 0 | | Multiplier input from internal row buffers. |
| 1 | | Multiplier input from external buffers. |
| **2 BIT 1** | | **FUNCTION = INPUT DATA DELAY** |
| 0 | 0 | No Data Delay Registers used. |
| 0 | 1 | One Data Delay Register used. |
| 1 | 0 | Two Data Delay Registers used. |
| 1 | 1 | Three data Delay Registers used. |
| **BIT 3** | | **FUNCTION = INPUT DATA FORMAT** |
| 0 | | Unsigned integer format. |
| 1 | | Two's complement format. |
| **BIT 4** | | **FUNCTION = COEFFICIENT DATA FORMAT** |
| 0 | | Unsigned integer format. |
| 1 | | Two's complement format. |
| **6 BIT 5** | | **FUNCTION = OUTPUT ROUNDING** |
| 0 | 0 | No rounding. |
| 0 | 1 | Round to 16 bits (i.e., DOUT19-4). |
| 1 | 0 | Round to 8 bits (i.e., DOUT19-12). |
| 1 | 1 | Not Valid. |
| **8 BIT 7** | | **FUNCTION = CASI0-15 INPUT SHIFT** |
| 0 | 0 | No shift. |
| 0 | 1 | Shift CASI0-15 left two. |
| 1 | 0 | Shift CASI0-15 left four. |
| 1 | 1 | Shift CASI0-15 left eight. |

### Row Length Register

The Row Length Register is used to store the programmed number of delays for the internal row buffers. The Programmed delay is set equal to the row length (r) of the input image. The input pixel data is stored in the row buffers to allow corresponding pixels of adjacent rows to be synchronously sent to the multiplier array for the convolution operation. The Row Length Register is programmable with the values from 0 to 1023, with 0 defined as a row length of 1024. Row lengths of 1 or 2 lead to meaningless results for a 3 x 3 kernel convolution, while a row length of 3 define 1 x 9 filter (See Operation Section). The Row Length Register is written through the use of A0-2, $\overline{CS}$ and $\overline{LD}$. Once the Row Length Register has been loaded, the convolver must reset before a new row length can be entered, or else new value will be ignored. After $\overline{RESET}$ returns high, user has 1024 cycles of CLK to load the Row Length Register. After 1024 CLK cycles, the Row Length Register is automatically set to 0 (row length = 1024) and further writes to this register are ignored.

### Coefficient Registers (CREG0, CREG1)

The control logic contains two Coefficient Register banks CR EG0 and CREG1. Each of these register banks is capable of storing nine 8-bit filter coefficient values (3 x 3 Kernel). The output of the registers are connected to the coefficient input of the corresponding multiplier in the 3 x 3 multiplier array (designated A through I). The register bank to be used for the convolution is selectable by writing to the appropriate address (See address decoder). All registers in a given bank are enabled simultaneously, and one of the banks is always active.

For most applications, only one of the register banks is necessary. The user can simply load CREG0 after power up, and use it for the entire convolution operation. (CREG0 is the Default Register). The alternate register bank allows the user to maintain two sets of filter coefficients and switch between them in real time. The coefficient masks are loaded via the CIN bus by using A0-2, $\overline{CS}$ and $\overline{LD}$. The selection of the particular register bank to be used in processing is also done by writing to the appropriate address (see address decoder). For example, if CREG0 is being used to provide coefficients to the multipliers, CREG1 can be updated at a low rate by an external processor; then at the proper time, CREG1 can be selected, so that the new coefficient mask is used to process the data. Thus, no clock cycles have been lost when changing between alternate 3 x 3 filter kernels.

The nine coefficients must be loaded sequentially over the CIN0-7 bus from A to I. The address of CREG0 or CREG1 is placed on A0-2, and then the nine coefficients are written to the corresponding Coefficient Register one at a time by using the $\overline{CS}$ and $\overline{LD}$ inputs.

### Address Decoder

The address decoder (see Figure 1) is used for writing to the control logic of the HSP48908. Loading an Internal Register is done by selecting the Destination Register with the A0-2 address lines, placing the data on CIN0-9, asserting the $\overline{CS}$ and $\overline{LD}$ control lines. When either $\overline{CS}$ or $\overline{LD}$ goes high, the data on the CIN0-9 lines is latched into the Addressed Register. The address map for the A0-2 bus is shown in Table 4.

While loading of the Control Logic Registers is asynchronous to CLK, the Target Register in the control logic is being read synchronous to the internal clock. Therefore, care must be taken when modifying the convolver setup parameters during processing to avoid changing the contents of the registers near a rising edge of CLK. The required setup time relative to CLK is given by the Specification TLCS. For example, in order to change the active Coefficient Register from CREG0 to CREG1 during an active convolution operation, a write will be performed to the address for selecting CREG1 for internal processing (A2 -0 = 110). In order to provide proper uninterrupted operation, $\overline{\text{LD}}$ should be deasserted at least TLCS prior to the next rising edge of CLK. Failure to meet this setup time may result in unpredictable results on the output of the convolver for one clock cycle. Keep in mind that this requirement applies only to the case where changes are being made in the control logic during an active convolution operation. In a typical convolver configuration routine, this specification would not be applicable.

**TABLE 4. ADDRESS MAP**

| CONTROL LOGIC ADDRESS MAP | |
|---|---|
| **A2-0** | **FUNCTION** |
| 000 | Load Row Length Register (RLR). |
| 001 | Load ALU Microcode Register (AMC). |
| 010 | Load Coefficient Register 0 (CREG0). |
| 011 | Load Coefficient Register 1 (CREG1). |
| 100 | Load Initialization Register (INT). |
| 101 | Select CREG0 for Internal Processing. |
| 110 | Select CREG1 for Internal Processing. |
| 111 | No Operation. |

## Cascade I/O

### Cascade Input

The cascade input lines (CASI0-15) have two primary functions. The first is used to allow convolutions with kernel sizes larger than 3 x 3. This can be implemented by connecting the DOUT bus of one convolver to the cascade inputs of another. The second function is for convolution on images wider than 1024 pixels. This type of operation can be implemented by using external row buffers to supply the pixel input data to the CASI0-15 inputs. The cascade input functions are determined by Initialization Register bit 0. When this bit is set to a “0”, the cascade input data is added to the convolver output. In this manner, multiple convolvers can be used to implement larger kernel convolution. When Initialization Register bit 0 is “1”, the data on CASI0-15 is divided into two 8-bit portions and is sent to the 3 x 3 multiplier array (refer to Block Diagram). This mode of operation allows the use of external row buffers for convolution of images with row sizes larger than 1024. Examples of these configurations are given in the Operations Section of this specification.

The data on the cascade inputs (CASI0-15) can also be left shifted by 0, 2, 4, or 8 bits. The amount of shift is determined by bits 7 and 8 of the Initialization Register (See Table 3). CASI0-15 is shifted by the specified number of bits and is added to the 20-bit output DOUT 0-19. The shifting function provides a method for cascading multiple HSP48908s and allowing a selectable amount of output growth while maximizing the resolution of the convolver result.

The cascade inputs can also be used as a simple way to add an offset to the convolved image. Bit 0 of the Configuration Register would be set to ‘0’, and the desired offset placed on the CASI0-15 inputs. While multiple offsets can be used and changed during the convolution operation, note that the required data setup and hold times with respect to CLK (TDS and TDH) must be met.

### Cascade Output

The cascade output lines (CASO0-7) are outputs from the second row buffer. Data at these outputs is the input pixel data delayed by two times the preprogrammed value in the Row Length Register. The cascade outputs are used to cascade multiple convolvers by connecting the cascade outputs of one device to the data inputs of another (see Operation Section).

## Control Signals

### HOLD

The HOLD control input provides the ability to disable internal clock and stop all operations temporarily. HOLD is sampled on the rising edge of CLK and takes effect during the following clock cycle (refer to Figure 2). This signal can be used to momentarily ignore data at the input of the convolver while maintaining its current output data and operational state.

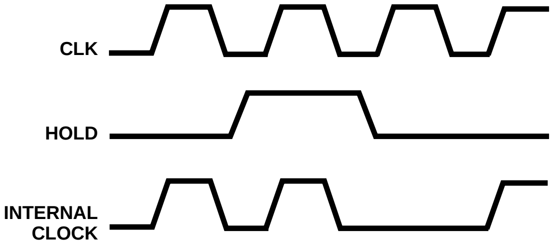


**FIGURE 2. HOLD OPERATION**

### $\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ signal initializes all internal flip flops and registers in the HSP48908. It is an asynchronous signal, and the convolver will remain in the reset state as long as $\overline{\text{RESET}}$ is asserted. On reset, all internal registers are set to zero or their default values, and all outputs are forced low. Following a reset, the default values in the internal registers will define the following mode of operation: internal row buffers used, line length = 1024, no input data delay, logical

A operation: output of ALU = A input (DIN0-7) output rounding and unsigned input data format

The convolver can be reset at any time, but must be reset before updating the Row Length Register in order to provide proper operation. After $\overline{\text{RESET}}$ returns high, the user has 1024 cycles of CLK to load the Row Length Register. After 1024 OLK cycles, the Row Length Register is automatically set to 0 (row length = 1024) and further writes to this register are ignored.

### $\overline{\text{FRAME}}$

This $\overline{\text{FRAME}}$ input initializes all internal flip flops and registers except for the coefficient, ALU, ALU microcode, Row Length, and Initialization Registers. It is used to reset the convolver between video frames and eliminates the need to reinitialize the entire convolver or reload the coefficients. $\overline{\text{FRAME}}$ is an asynchronous input and may occur at any time. However, it must be deasserted at least TFS ns prior to the rising clock edge that is to begin operation for the next frame. While $\overline{\text{FRAME}}$ is asserted, the registers and flip- flops will remain in the reset state.

## Operation

The HSP48908 has three basic modes of operation: single chip mode, operation with external row buffers and multiple devices cascaded together for larger convolution kernels and/or longer row lengths. The mode of operation is defined by the contents of the Initialization Register, and can be modified at any time by a microprocessor or other external means.

### Single Chip Mode

A single HSP48908 can be used to perform 3 x 3 convolution on 8-bit image data with row lengths up to 1024. A block diagram of this configuration is shown in Figure 3. In this mode of operation, the image data is input into the DIN0-7 bus in a raster scan order starting with the upper left pixel. To perform the convolution operation, a group of nine image pixels is multiplied by the 3 x 3 array of filter coefficients and their products are summed and sent to the output. For the example in Figure 3, the pixel value in the output image at location (m, n) is given by:

POUT(m,n) (A x Pm-1, n-1) + (B x Pm-1, n) + (C x Pm-1, n+1)
+ (D x Pm, n-1) + (E x Pm, n) + (F x Pm, n+1)
+ (G x Pm+1, n-1) + (H x Pm+1, n)+ (I x Pm+1, n+1)

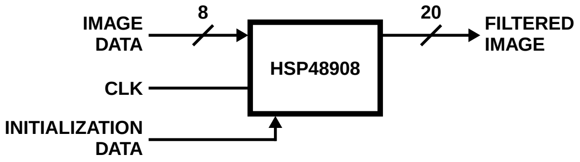
This process is continually repeated until the last pixel of the last row of the image has been input. It can then start again with the first row of the next frame. The $\overline{\text{FRAME}}$ pin is used to clear the row buffers, multiplier input latches and DOUTO19 registers between frames.

The setup for single chip operation is straightforward. After reset, the convolver is configured for row lengths of 1024 pixels, no input data delay, no ALU pixel point operations, no output rounding, and an unsigned input format. The user can change this default setup by loading new values into the ALU microcode, initialization and Row Length Registers. $\overline{\text{RESET}}$ also clears the Coefficient Registers and CREG0 is selected for internal processing. The user can now load the coefficients one at a time from A to I, via the CIN0-7 inputs and the $\overline{\text{LD}}$ and $\overline{\text{CS}}$ control lines.

Multiple filter kernels can also be used on the same image data using the dual Coefficient Registers CREG0 and CREG1. This type of filtering is used when the characteristics of the input pixel data change over the image in such a way that no single filter produces satisfactory results for the entire image. In order to filter such an image, the characteristics of the filter itself must change while the image is being processed. The HSP48908 can perform this function with the use of an external processor. The processor is used to calculate the required new filter coefficients, loads them into the Coefficient Register not in use, and selects the newly loaded Coefficient Register at the proper time. The first Coefficient Register can then be loaded with new coefficients in preparation for the next change. This can be carried out with no interruption in processing, provided that the new register is selected synchronous to the convolver CLK signal.

The HSP48908 can also operate as a one dimensional 9 tap FIR filter by programming the Row Buffer Length Register with a value of 3 and setting the Initialization Register bit-0 to a “0”. This configuration will provide for nine sequential input values in the input to be multiplied by the coefficient values in the selected Coefficient Register and provide the proper filtered output. The equation for the output then becomes:

$$\begin{aligned} D_{OUTn} = {} & A \times Dn_{-8} + B \times Dn_{-7} + C \times D_{n-6} + D \times D_{n-5} \\ & + E \times D_{n-4} + F \times D_{n-3} + G \times D_{n-2} + H \times D_{n-1} \\ & + I \times D_{n} \end{aligned}$$



| FILTER KERNEL | IMAGE DATA | | |
|---|---|---|---|
| A B C | Pm-1, n -1 | Pm-1, n | Pm-1, n + 1 |
| D E F | PM, n -1 | Pm, n | Pm, n + 1 |
| G H I | Pm + 1, n -1 | Pm + 1, n | Pm + 1, n + 1 |

**FIGURE 3. 3 x 3 KERNEL ON AN 8-BIT, 1024 x N IMAGE**

### Use Of External Row Buffers

External row buffers may be used when frames with row sizes larger than 1024 pixels are desired. To use the HSP48908 in this mode, the cascade mode control bit (bit 0) of the Initialization Register is set to ‘1’ to allow the data on the cascade inputs CASI0-15 to go to the multiplier array. The inputs of one external row buffer (such as the HSP9500) are connected to the input data in parallel with the DIN0-7 lines of the convolver; and its outputs are connected to the CASI0-7 inputs (See Figure 4). A second external row buffer is connected between the outputs of the first row buffer and the CASI8-15 inputs of the convolver. The convolution operation can then be performed by the HSP48908 in the same manner as the single chip mode. The row length in this configuration is limited only by the maximum length of the external row buffers. Note that when using the convolver in this configuration, the programmable input data delays and ALU will only operate on the data entering the DIN0-7 inputs (i.e., the bottom row of the 3 x 3 sum of products). If higher order filters or pixel point operations are required when using external row buffers, these functions must be implemented externally by the user.

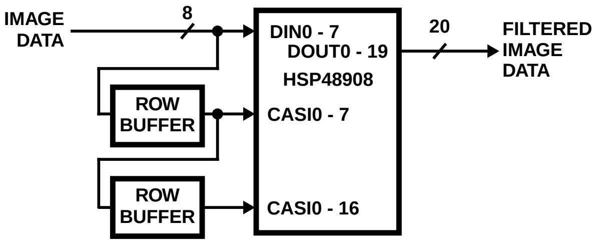


FIGURE 4. USING EXTERNAL ROW BUFFERS WITH THE HSP48908

### Cascading Multiple HSP48908s

Multiple HSP48908s are capable of being cascaded to perform convolution on images with row lengths longer than 1024 pixels and with kernel sizes larger than 3 x 3. Figure 5 illustrates the use of two HSP48908s to perform a 3 x 3 kernel convolution on a 2K x N frame. In this case, the cascade mode control bit (Bit 0) of both Initialization Registers are set to a ‘0’. The loading of the coefficients is accomplished just as before. However, the 3 x 3 mask is divided into two portions for proper convolution output as follows: Convolver #1 = DEF000GHI and Convolver #2 = ABC000000.

| 3 x 3 FILTER KERNEL | COEFFICIENT MASKS | |
|---|---|---|
| | CONVOLVER #1 | CONVOLVER #2 |
| A B C | D E F | A B C |
| D E F | 0 0 0 | 0 0 0 |
| G H I | G H I | 0 0 0 |

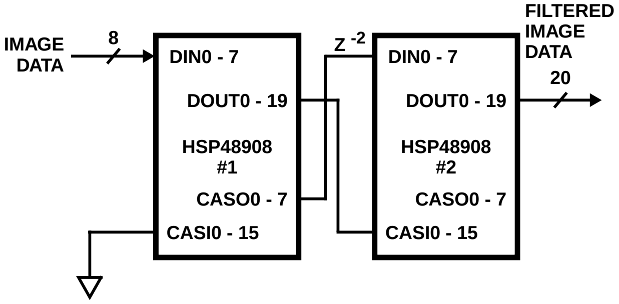
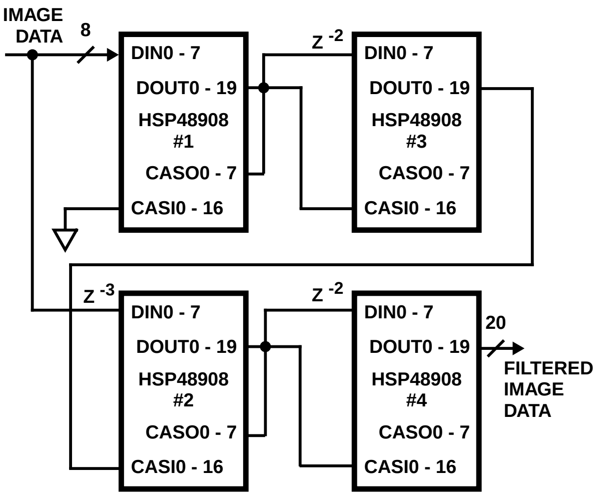


FIGURE 5. 3 x 3 KERNAL CONVOLUTION ON A 2K x N IMAGE

The same configuration can be used to perform 3 x 5 convolution on a 1K x N frame simply by setting up the coefficients of the convolvers to implement the 3 x 5 mask as indicated below:

| 3 x 5 FILTER KERNEL | CONVOLVER COEFFICIENT MASKS | |
|---|---|---|
| A B C | G H I | A B C |
| D E F | J K L | D E F |
| G H I | M N O | 0 0 0 |
| J K L | | |
| M N O | | |

In addition to larger frames, larger kernels can also be addressed through cascadability. An example of the configuration for a 5 x 5 kernel convolution on a 1K x N frame is shown in Figure 6. Note that in this configuration, convolver #2 incorporates a 3 clock cycle delay (z -3) and convolvers 3 and 4 incorporate 2 clock cycle delays (z -2) at their pixel inputs. These delays are required to ensure proper data alignment in the final sum of products output of the cascaded convolvers. The number of delays required at the pixel input is programmable through the use of bits 1 and 2 of the Initialization Register (Refer to Table 3).



| 5 x 5 FILTER KERNEL | CONVOLVER COEFFICIENT MASKS | |
|---|---|---|
| A B C D E | O K L | O A B |
| F G H I J | O P Q | O F G |
| K L M N O | O U V | 0 0 0 |
| P Q R S T | | |
| U V W X Y | M N O | C D E |
| | R S T | H I J |
| | W X Y | 0 0 0 |

FIGURE 6. 5 x 5 KERNEL CONVOLUTION ON A 1K x N IMAGE

In any of the cascade configurations, only 16 bits of the 20-bit output (DOUT0-19) can be connected to the 16 cascade inputs (CASI0-15) of another convolver. Which 16 bits are chosen, depends upon the amount of growth expected at the convolver output. The amount of growth is dependent on the input pixel data and the coefficients selected for the convolution operation. The maximum possible growth is calculated in advance by the user, and the convolvers are set up to appropriately shift the cascade input data through the use of bits 7 and 8 of the Initialization Register (see Cascade I/O). Referring to Figure 6, if the maximum growth out of convolver #1 extends into bit 16 or 17, then DOUT2-17 is connected to the cascade inputs of convolver #3, which is programmed to shift the input data left by two bits. Likewise, if the data out of convolver #3 grows into bit 18 or 19, then DOUT4-19 are connected to the CASI0-15 inputs of convolver #2, which is programmed to shift the input left by 4 bits.

### Cascading For Row Sizes Larger Than 1024

Combining large images with large kernels is accomplished by implementing external row buffers, external Data Delay Registers and external adders. Figure 7 illustrates a circuit for implementation of a 5 x 5 convolution on a 2K x N image. The 5 x 5 coefficient mask is again distributed among the four HSP48908's. The width of the DOUT path to be used in this case is dependent on the amount of resolution required and the amount of growth expected at the output.

### Frame Rate

The total time to process an image is given by the formula:

T = R x C/F, where:

T = time to process a frame.

R = number of rows in the image.

C = number of pixels in a row.

F = clock rate of the HSP48908.

Note that the size of the kernel does not enter into the equation. Convolvers cascaded for larger kernels or larger frame sizes, as in the examples shown, process the image in the same amount of time as a single HSP48908 convolving the image with a 3 x 3 kernel. Therefore, there is no performance degradation when cascading multiple HSP48908s.

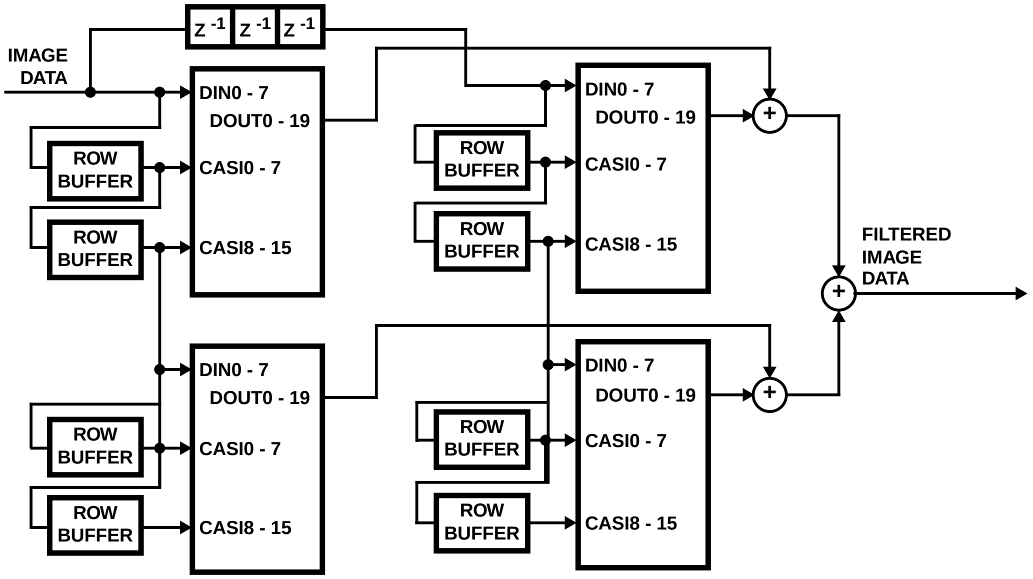


FIGURE 7. 5 x 5 KERNEL CONVOLUTION ON A 2K x N IMAGE

## Absolute Maximum Ratings

| | |
|---|---|
| Supply Voltage | +8.0V |
| Input, Output or I/O Voltage Applied | GND -0.5V to $V_{CC}$ +0.5V |
| ESD Classification | Class 1 |

## Operating Conditions

| | |
|---|---|
| Temperature Range | $0^oC$ to $70^oC$ |
| Voltage Range | +4.75V to 5.25V |

## Thermal Information

| Thermal Resistance (Typical, Note 1) | $\theta_{JA}$ ($^oC/W$) | $\theta_{JC}$ ($^oC/W$) |
|---|---|---|
| MQFP Package | 48.0 | N/A |
| PLCC Package | 34.0 | N/A |
| PGA Package | 35.0 | 6.0 |

| | |
|---|---|
| Maximum Junction Temperature ($T_J$) | |
| MQFP Package | $150^oC$ |
| PLCC Package | $150^oC$ |
| PGA Package | $175^oC$ |
| Maximum Storage Temperature Range | $-65^oC$ to $150^oC$ |
| Maximum Lead Temperature (Soldering 10s) | $300^oC$ |
| (PLCC, MQFP - Lead Tips Only) | |

## Die Characteristics

| | |
|---|---|
| Number of Transistors or Gates | 190,000 Transistors |

*CAUTION: Stresses above those listed in “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.*

NOTE:

1. $\theta_{JA}$ is measured with the component mounted on an evaluation PC board in free air.

## DC Electrical Specifications

$V_{CC}$ = 5.0V + 5%, $T_A$ = $0^oC$ to $70^oC$

| PARAMETER | SYMBOL | TEST CONDITIONS | MIN | MAX | UNITS |
|---|---|---|---|---|---|
| Logical One Input Voltage | $V_{IH}$ | $V_{CC}$ = 5.25V | 2.0 | - | V |
| Logical Zero Input Voltage | $V_{IL}$ | $V_{CC}$ = 4.75V | - | 0.8 | V |
| High Level Clock Input | $V_{IHC}$ | $V_{CC}$ = 5.25V | 3.0 | - | V |
| Low Level Clock Input | $V_{ILC}$ | $V_{CC}$ = 4.75V | - | 0.8 | V |
| Output HIGH Voltage | $V_{OH}$ | $I_{OH}$ = 400µA, $V_{CC}$ = 4.75V | 2.6 | - | V |
| Output LOW Voltage | $V_{OL}$ | $I_{OL}$ = +2.0mA, $V_{CC}$ = 4.75V | - | 0.4 | V |
| Input Leakage Current | $I_I$ | $V_{IN}$ = $V_{CC}$ or GND, $V_{CC}$ = 5.25V | -10 | 10 | µA |
| I/O Leakage Current | $I_O$ | $V_{OUT}$ = $V_{CC}$ or GND | -10 | 10 | µA |
| Standby Power Supply Current | $I_{CCSB}$ | $V_{IN}$ = $V_{CC}$ or GND, $V_{CC}$ = 5.25V, Outputs Open | - | 500 | µA |
| Operating Power Supply Current | $I_{CCOP}$ | f = 20MHz, $V_{IN}$ = $V_{CC}$ or GND, (Note 2) | - | 140 | mA |
| Input Capacitance | $C_{IN}$ | f = 1MHz, $V_{CC}$ = Open, All Measurements are referenced to device GND (Note 3). | - | 10 | pF |
| Output Capacitance | $C_O$ | | - | 12 | pF |

NOTES:

2. Power supply current is proportional to operating frequency. Typical rating for $I_{CCOP}$ is 7.0mA/MHz.
3. Not tested, but characterized at initial design and at major process/design changes.

## AC Electrical Specifications

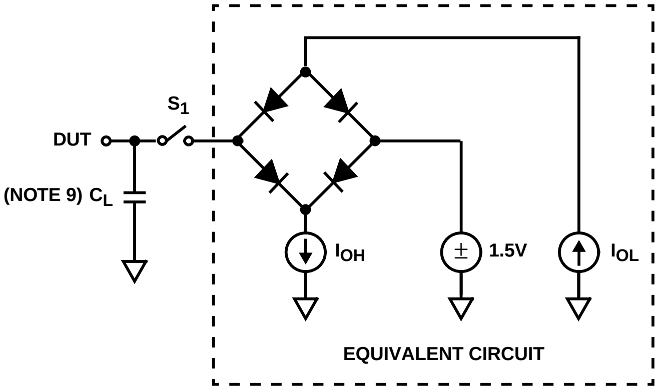
$V_{CC}$ = 5.0V ±5%, $T_A$ = $0^oC$ to $70^oC$

| PARAMETER | SYMBOL | NOTES | -32 (32MHz) | | -20 (20MHz) | | UNITS |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| Clock Period | $t_{CYCLE}$ | | 31 | - | 50 | - | ns |
| Clock Pulse Width High | $t_{PWH}$ | | 12 | - | 20 | - | ns |
| Clock Pulse Width Low | $t_{PWL}$ | | 13 | - | 20 | - | ns |
| Data Input Setup Time | $t_{DS}$ | | 13 | - | 14 | - | ns |
| Data Input Hold Time | $t_{DH}$ | | 0 | - | 0 | - | ns |
| Clock to Data Out | $t_{OUT}$ | | - | 16 | - | 22 | ns |
| Address Setup Time | $t_{AS}$ | | 13 | - | 13 | - | ns |
| Address Hold Time | $t_{AH}$ | | 0 | - | 0 | - | ns |
| Configuration Data Setup Time | $t_{CDS}$ | | 14 | - | 16 | - | ns |
| Configuration Data Hold Time | $t_{CDH}$ | | 0 | - | 0 | - | ns |
| $\overline{LD}$ Pulse Width | $t_{LPW}$ | | 12 | - | 20 | - | ns |
| $\overline{LD}$ Setup Time | $t_{LCS}$ | Note 4 | 25 | - | 30 | - | ns |
| CIN0-7 Setup to CLK | $t_{CS}$ | | 14 | - | 16 | - | ns |
| $\overline{CS}$ Setup to $\overline{LD}$ | $t_{CSS}$ | | 0 | - | 0 | - | ns |
| CIN0-7 Hold Time from CLK | $t_{CH}$ | | 0 | - | 0 | - | ns |
| $\overline{CS}$ Hold from $\overline{LD}$ | $t_{CSH}$ | | 0 | - | 0 | - | ns |
| $\overline{RESET}$ Pulse Width | $t_{RPW}$ | | 31 | - | 50 | - | ns |
| $\overline{FRAME}$ Setup to Clock | $t_{FS}$ | Note 5 | 21 | - | 25 | - | ns |
| $\overline{FRAME}$ Pulse Width | $t_{FPW}$ | | 31 | - | 50 | - | ns |
| EALU Setup Time | $t_{ES}$ | | 12 | - | 14 | - | ns |
| EALU Hold Time | $t_{EH}$ | | 0 | - | 0 | - | ns |
| HOLD Setup Time | $t_{HS}$ | | 11 | - | 12 | - | ns |
| HOLD Hold Time | $t_{HH}$ | | 1 | - | 1 | - | ns |
| Output Enable Time | $t_{EN}$ | Note 6 | - | 16 | - | 22 | ns |
| Output Disable Time | $t_{OZ}$ | Note 8 | - | 28 | - | 32 | ns |
| Output Rise Time | $t_R$ | From 0.8V to 2.0V, Note 8 | - | 6 | - | 6 | ns |
| Output Fall Time | $t_F$ | From 2.0V to 0.8V, Note 8 | - | 6 | - | 6 | ns |

NOTES:

4. This specification applies only to the case where the HSP48908 is being written to during an active convolution cycle. It must be met in order to achieve predictable results at the next rising clock edge. In most applications, the configuration data and coefficients are loaded asynchronously and the $T_{LCS}$ Specification may be disregarded.
5. While $\overline{FRAME}$ is an asynchronous signal, it must be deasserted a minimum of $T_{FS}$ ns prior to the rising clock edge which is to begin loading pixel data for a new frame.
6. Transition is measured at ±200mV from steady state voltage with loading as specified in test load circuit with $C_L$ = 40pF.
7. AC Testing is performed as follows: Input levels (CLK Input) 4.0 and 0V, Input levels (all other Inputs) 0V and 3.0V, Timing reference levels (CLK) = 2.0V, (Others) = 1.5V; output load per test load circuit with $C_L$ = 40pF. Output transition is measured at $V_{OH} \geq 1.5V$ and $V_{OL} \leq 1.5V$.
8. Controlled via design or process parameters and not directly tested. Characterized upon initial design and after major process and/or design changes.

## Test Load Circuit



NOTES:

9. Includes stray and jig capacitance.
10. Switch $S_1$ Open for $I_{CCSB}$ and $I_{CCOP}$ Tests.

## Timing Waveforms

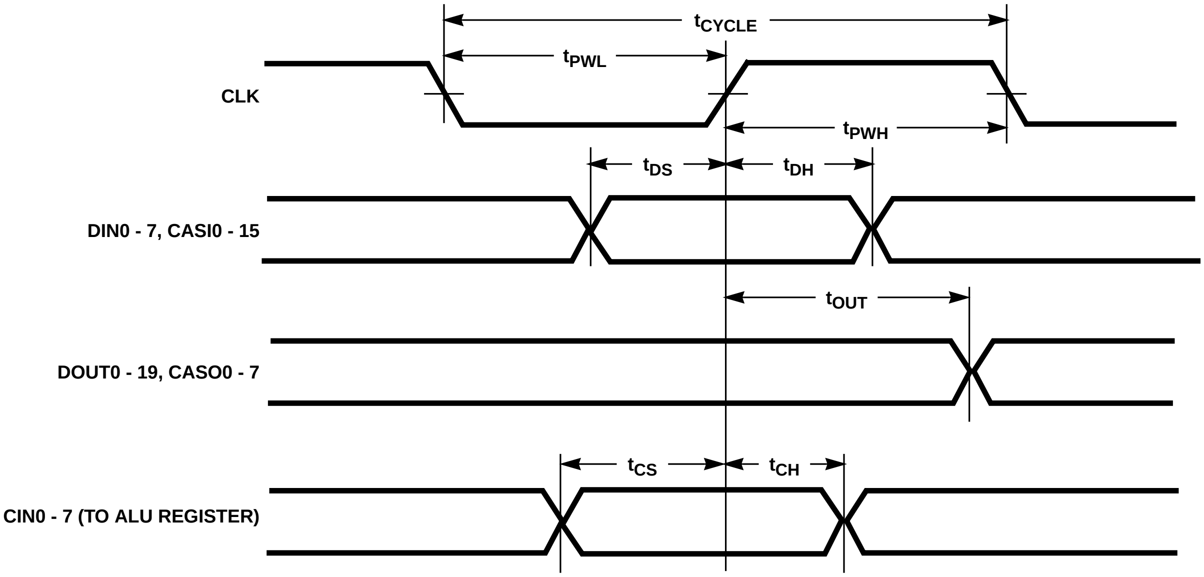
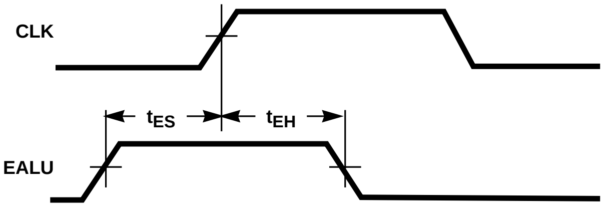


FIGURE 8. FUNCTIONAL TIMING



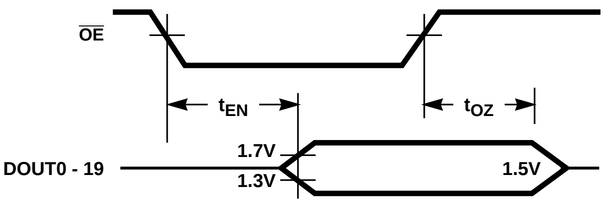


FIGURE 9. EALU TIMING

FIGURE 10. THREE-STATE CONTROL

## Timing Waveforms (Continued)

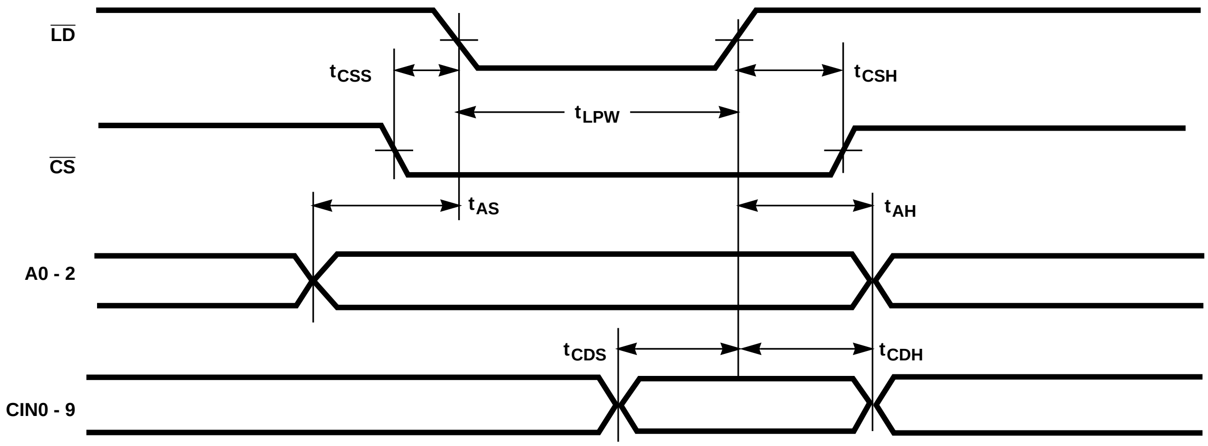


FIGURE 11. CONFIGURATION TIMING

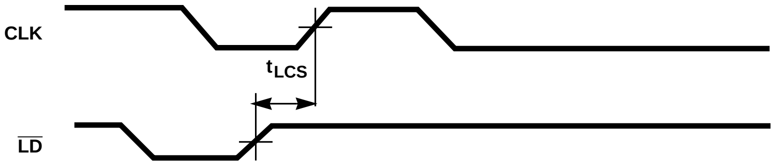


FIGURE 12. SYNCHRONOUS LOAD TIMING

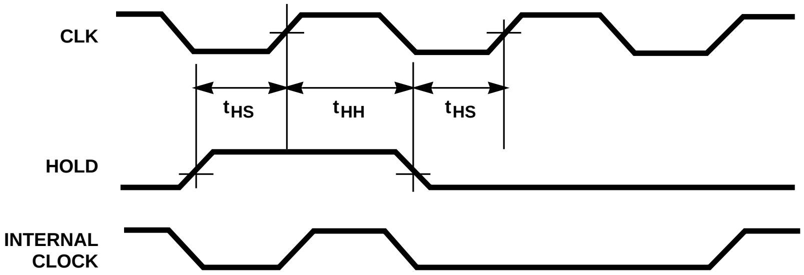


FIGURE 13. HOLD TIMING

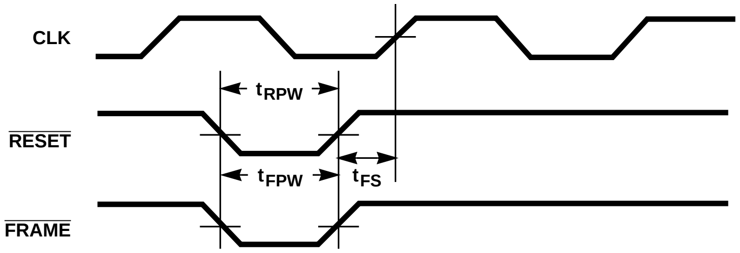


FIGURE 14. $\overline{\text{FRAME}}$ AND $\overline{\text{RESET}}$ TIMING

All Intersil semiconductor products are manufactured, assembled and tested under **ISO9000** quality systems certification.

*Intersil semiconductor products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.*

For information regarding Intersil Corporation and its products, see web site **http://www.intersil.com**