AccelWare IP cores provide a direct path to hardware implementation for complex MATLAB® toolbox and built-in functions. AccelWare cores deliver synthesizable, pre-verified DSP functions that enable true, top-down MATLAB architectural synthesis of FPGAs and ASICs. AccelWare IP includes Building Block, Advanced Math, Signal Processing and Communications Toolkits.

Matrix Inverse (Cholesky method)

The Matrix Inverse with Cholesky method core computes the inverse of a symmetric, positive definite input matrix. The implementation of the inverse computation is based on the Cholesky factorization of the input matrix followed by a product of the inverse of the matrix factors. (For applications where only matrix factorization is required, see the data sheet for the **Matrix Factorization – Cholesky method**).

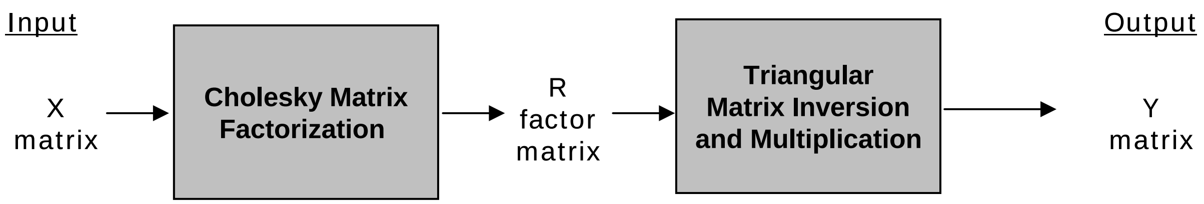

Figure 1: Matrix Inverse Block Diagram

The Matrix Inverse model uses a factorization algorithm based on a column-wise version of the Cholesky factorization algorithm described in Golub and Van Loan [1].

Input			
Signal Name	**Signal Description**	**Type**	**Range**
X	Input matrix to be factored	Real	4 to 24 bits (fixed-point representation)

Output			
Signal Name	**Signal Description**	**Type**	**Range**
Y	Output inverse matrix	Real	Up to 32 bits (fixed-point representation)

[1] G. H. Golub, C. F. Van Loan, "Matrix Computations", The Johns Hopkins University Press, Baltimore, Maryland, 1996.

Implementation Parameters		
Parameter Name	**Parameter Description**	**Range**
X quantizer	Input matrix quantization	• Mode – fixed/ufixed • Round Mode – floor • Overflow Mode – wrap/saturate • Word Length – 2 to 24 • Fraction Length – 0 to Word Length (or Word Length – 1) if Mode parameter is set to ufixed (or fixed).
Input Data Type	Input matrix data type	Real
Matrix Size	Input matrix size	Integer in range 3 to 64
Input/Output Type	Input/output matrix dimension representation	• 1-D • 2-D
Output Precision	Number of bits for output matrix	Auto – sets value based on input matrix word-length.
Resource Sharing	Resource-shared implementation option	Yes

Input Matrix Quantization

The number representation of the input is defined by the input matrix quantization. The *chol_inverse* accepts real-valued, fixed-point input data with quantization properties defined by this parameter.

Input Data Type

The input matrix must be real-valued. The input matrix is assumed to be symmetric and positive definite.

Matrix Size

The matrix size is defined by the Matrix Size parameter.

Input/Output Data Type

The *chol_inverse* model can be generated to accept input and generate output matrices as 1-D or 2-D arrays.

Output Precision

This parameter defines the precision, in number of bits, of the output matrix.

Hardware Interfacing

A synthesizable AccelWare MATLAB model will typically be a design module that is part of a larger design on a chip. The flow of data into and out of the hardware ports is controlled by a protocol called DAP (Data Accept Protocol). Synthesizing the *chol_inverse* model with AccelChip in a stand-alone fashion will produce a Matrix Inverse hardware block with DAP interface signals ready for integration into a larger system. The following gives a description of DAP interface protocol.

Global Signals

The hardware module has one Clock input and one global Reset. Data transfers on each port are synchronized to the Clock. The global Reset returns all registers and flip-flops to a known state.

Input Synchronization Signals

ND (NewData) -This signal is controlled by the external design and indicates that data on the input data bus is valid. This causes the receiving device (the hardware module) to capture the data on the rising edge of the next clock cycle.

RFND (ReadyForNewData) -This signal is controlled by the hardware module and indicates that the module is ready to capture new data from the input bus. When the module sets RFND low, the external design should immediately stop sending new data. If the hardware module holds RFND constantly high, then new data will be captured on every clock cycle provided the sending device can send data that fast.

Output Synchronization Signals

Done -This signal is controlled by the hardware module and indicates that data on the output data bus is valid. Once Done is set high, it will remain high until the receiving device acknowledges the data capture by setting DA high. If the external design holds DA constantly high, then the hardware module will send data at the maximum possible rate, as governed by the module clock frequency and the latency of the computing algorithm.

DA (Data Accept) - This signal is controlled by the external design and indicates that the data on the output bus has been captured. If the external design holds DA constantly high, then the hardware module will send data out at the maximum rate possible, as governed by the module clock frequency and the latency of the computing algorithm.

DA (Data Accept) - This signal is controlled by the external design and indicates that the data on the output bus has been captured. If the external design holds DA constantly high, then the hardware module will send data out at the maximum rate possible, as governed by the module clock frequency and the latency of the computing algorithm.

Signal	Direction	Description
Clock	Input	Clock input
Reset	Input	Reset input
X_flat	Input	X Input matrix data
RFND_X	Output	X Ready for new data
ND_X	Input	X New input data valid
Y_write_flat	Output	Y Output matrix data
Done_Y_write	Output	Y Done indication
DA_Y_write	Input	Y Data accepted indication

Figure 2: DAP Signals in *chol_inverse*

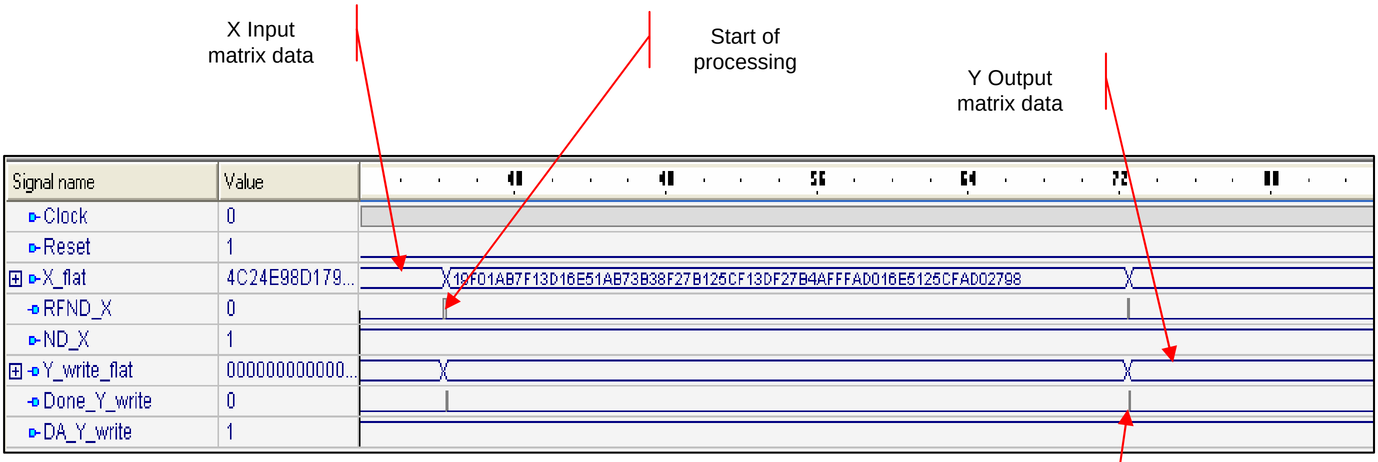

Figure 4: Input/Output Timing Example

Differences in Operation between AccelWare *chol_inverse()* and MATLAB *inv()*

The MATLAB *inv* function can operate on asymmetric, non-positive definite matrices. The AccelWare *chol_inverse* is constrained to operate on symmetric, positive definite matrices due to the applicability of Cholesky factorization.

The MATLAB *inv* function can operate on matrices with complex data. The AccelWare *chol_inverse* is designed to operate directly on real-valued matrices only.

Ordering Information

The AccelWare *chol_inverse* core is included in the AccelWare Advanced Math Toolkit (AccelChip part number **AWAMT**) and is available as an option to the AccelChip DSP Synthesis product (AccelChip part number **ACDSP**).

For further information on availability, contact your local AccelChip sales representative or send email to sales@accelchip.com.

AccelChip Incorporated
1900 McCarthy Blvd., Suite. 204, Milpitas, CA 95035 phone (408) 943 0700 option 1 fax (408) 943 0661