ImageFlow

Imaging Applications Development

Datacube's pipeline architecture provides unmatched power to meet the challenge of processing digital images at frame rates. The complex process of controlling multiple parallel pipelines of data is accomplished with Datacube's powerful ImageFlow software.

ImageFlow, a library of C-callable functions, configures and manages data transfers and processing elements on Datacube's family of MaxVideo pipeline processing devices. It accomplishes this task with minimal overhead, acheiving near-register-level performance without painstaking effort. It significantly simplifies pipeline processing control, handling complex synchronization and timing issues for the programmer. ImageFlow is proven and robust, having been continuously improved through more than five years of development, field testing, and user feedback.

Pipeline Processing

Pipeline processing uses a collection of specialized computational elements connected sequentially. Each of these elements is designed to optimize the performance of a specific image processing task.

With Datacube's pipeline processors, 2-D images are converted to 1-D streams of video data and piped through a configurable series of these specialized computational elements. Elements can operate simultaneously to produce a steady stream of processed data. Multiple parallel pipelines work synchronously to further enhance data throughput.

With ImageFlow, the computational elements can be quickly reconfigured to perform different sets of imaging operations. These operations are set up prior to the flow of data through the pipes. Once the pipeline processor begins operations, reconfiguration can occur "on-the-fly" between frames, providing exceptional flexibility and power.

Elements Model

ImageFlow is based on Datacube's comprehensive image processing elements model. This model formally characterizes pipeline elements in a way that provides all the information required by the programmer, while automatically attending to the details that are specific to a particular hardware implementation. All Datacube hardware is modeled as a collection of these elements. As new elements are developed, the model is expanded to include them.

The elements are hierarchally classified by function, such as ALUs, LUTs, delays, timing elements, and data port elements, including MAXbus, generic digital, and analog ports.

Three elements play particularly important roles: *Surface Stores* are rectangular arrays that store image data that is accessed and transferred at a later time; *Gateways* are the elements at the beginning and end of pipes where image data is converted between 2-D surface stores and 1-D streams that flow through the pipes; and *Processing Elements* are the specialized computational elements within the pipes that perform operations on the image data.

Application Development

The first step in application development is to define algorithms that effi-

- Optimizes the superb performance characteristics of MaxVideo image processing devices
- Greatly simplifies the most difficult aspects of pipeline processor programming (pipeline timing, etc.)
- Supports video acquisition from a wide range of cameras and sensors
- Exceptionally reliable through many years of development over several product generations
- Provides support for SunOS, Solaris, LynxOS, VxWorks, and other operating systems
- Comprehensive introductory and advanced training courses, and expert support available

ciently accomplish the desired image processing task. These algorithms are then mapped to a sequence of pipeline processor elements, or topology, for implementation.

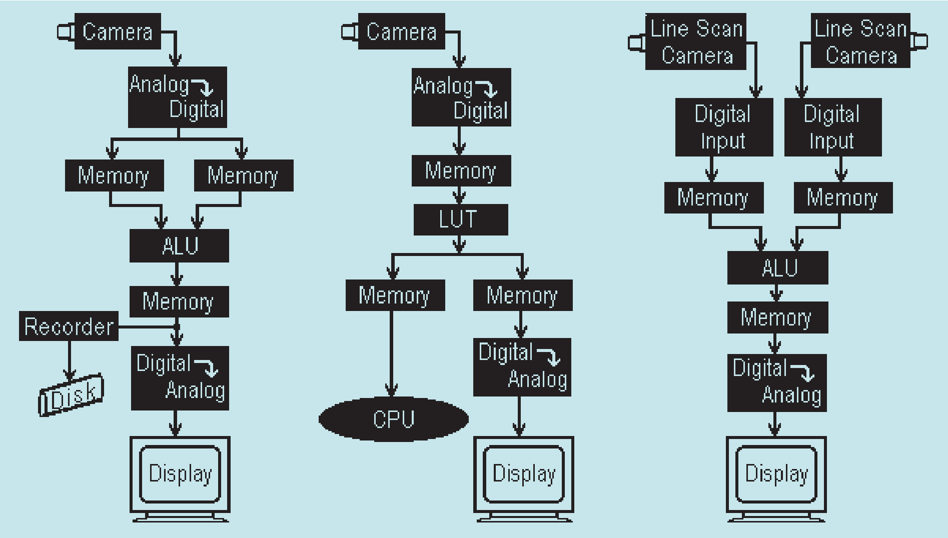
Pipe topologies can vary greatly. Multiple data sources, multiple data destinations, forking of data (into parallel paths), and merging of data (back into a single path) are all supported. Additionally, multiple parallel pipes can be run simultaneously.

ImageFlow element flow diagrams for each pipeline processor device provide a basis for algorithm mapping. These diagrams show all pipeline elements on the device, and all data pathways between them.

After the pipeline elements needed to implement the algorithm are defined, a processing pipeline is built. Calls are made to “connect functions” that set programmable switches, routing the data through the appropriate sequence of pipeline processor elements, and effectively tying multiple elements together into a processing pipeline.

Most applications require efficient allocation of processing resources. The goal, therefore, is to choose paths that contain the required elements, and include only those elements that are actually needed. Bypassed elements are then available to implement another portion of the algorithm, or another algorithm in a different pipe that can run simultaneously with the first.

Next, attribute-setting functions are called. These functions set the programmable attributes of the individual pipeline elements so that the elements in a pipe perform the desired processing operations (e.g., set an ALU element to add, select a LUT bank load with inverse transform, etc.).



ImageFlow allows a wide variety of multiple parallel pipeline architectures to be set up, controlled, and reconfigured on a frame-to-frame basis to optimize the processing of digital images.

Once the pipe has been built and the processing attributes have been set, the next step is to arm the pipe. Pipe arming, handled by the Data Stream Manager (DSM), automates the very complex process of calculating and setting all of the timing parameters of each pipeline. With data flowing in and/or out of image memories, and through pipe branches of different lengths, it is imperative to properly synchronize all data transfers. The DSM traces each pipe’s topology, determines the amount of delay in each branch of each pipe, accounts for the horizontal and vertical size of source and destination image data, and takes care of other details. It then performs a set of complex calculations (on the host machine) and sets each pipe’s timing parameters so that all data leaves from and/or arrives at each pipeline element at the proper instant in time.

Finally, the actual image processing operations are performed by firing data through each of the pipes. Data can be fired through a pipe as a single-shot transfer, or as a series of continuously occurring transfers.

Like any slave processor, these pipelines are themselves asynchronous processes. The independent pipeline processes, and the host process that manages the pipeline’s creation, must therefore be synchronized. The ImageFlow Event Manager supports process control of real-time transfers to provide the necessary synchronization.

Additional Information

For more information about the products mentioned in this document, please refer to the following Datacube literature:

ImageFlow Technical Description
MaxVideo 250 Data Sheet

ImageFlow, and MaxVideo 250 are trademarks of Datacube, Inc. All other trademarks are held by their respective owners. Specifications subject to change without notice. (11/98) DS0024-2.1

Major ImageFlow Components

- **IP Element Control Library:** Includes all functions that control attributes of the elements
- **Data Stream Manager:** Controls data transfers between all data surfaces through data pipelines
- **Event Manager:** Supports process control of real-time transfers
- **Configuration Utilities:** ASCII configuration file describes the configuration of all devices
- **Error Handler:** Provides centralized error-handling capabilities that simplify applications error testing
- **Device-Specific Functions:** Many devices have specialized functions
- **Graphic Functions:** Provides graphics, text and bit-block transfer for devices with image memory
- **Video Object Functions:** Offer a variety of acquisition modes to interface with most cameras and sensors

DATACUBE