Design of an event-driven random-access-windowing CCD-based camera

Steve P. Monacos*, Raymond K. Lam, Angel A. Portillo, Gerardo G. Ortiz
Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., Pasadena, CA 91109-8099, MS 300-123

ABSTRACT

Commercially available cameras are not designed for the combination of single frame and high-speed streaming digital video with real-time control of size and location of multiple regions-of-interest (ROI). A new control paradigm is defined to eliminate the tight coupling between the camera logic and the host controller. This functionality is achieved by defining the indivisible pixel read out operation on a per ROI basis with in-camera time keeping capability. This methodology provides a Random Access, Real-time, Event-driven (RARE) camera for adaptive camera control and is well suited for target tracking applications requiring autonomous control of multiple ROIs. This methodology additionally provides for reduced ROI read out time and higher frame rates compared to the original architecture by avoiding external control intervention during the ROI read out process.

Keywords: Region-of-interest, CCD, real-time, event-driven, adaptive, target-tracking

1 INTRODUCTION

Previously, we reported on a custom camera with real-time software control of single line readout for acquiring two ROIs[1]. This architecture required intimate coupling between the host controller and the camera logic with an associated penalty in terms of system operational speed and functional capabilities. In this paper we describe the functionality, architecture and control methodology of a Random Access, Real-time, Event-driven (RARE) camera as part of a real-time target acquisition and tracking platform. The camera implementation uses a Texas Instruments TC237 charge-coupled device (CCD) focal plane array (FPA) and two TLV987 signal processors. Control of the imager and signal processors is via custom logic that accepts user commands and provides region-of-interest (ROI) pixel data. This methodology results in a message-passing paradigm and provides real-time imager control without knowledge of detailed imager operation.

In the remainder of this section, we discuss the system functional goals and RARE camera requirements. Section 2 describes the RARE camera hardware architecture. Section 3 details the camera interface and software control. Section 4 presents experimental results of camera operation. Conclusions are presented in section 5.

1.1 Camera requirements

The primary motivation for this camera development is to realize an adaptive sensor mechanism as part of a platform for real-time autonomous acquisition and tracking applications[2]. Such a platform requires both a sensor and a control philosophy that provides real-time adaptation of the sensor based on target characteristics and dynamics and environmental conditions. To achieve this goal requires a camera capable of frame rates of several hundred to several thousand frames per second with operating parameters that can be adjusted on a per-frame basis.

High frame rates with adaptive imager control are achieved with a conventional CCD by reading out only the pixel regions of interest and discarding all other pixels. This mode of operation required the development of a customized local controller for the CCD imager to provide a tightly coupled mechanism for imager operation. Configuration of the controller is handled by a host tracking processor, discussed in section 2, that loads initialization and tracking parameters into the controller to define imager operation. The initialization parameters are needed for defining the start up mode of the controller and the tracking parameters define detailed operation of the imager during acquisition and tracking operations.

* steve.monacos@jpl.nasa.gov; phone 818 354-9161; fax 818 393-3302

The functional evolution of the RARE camera is shown in Table 1. Version 1.0 used software control of low-level CCD operations as a first implementation but required tight coupling of the camera with the host tracking processor. This release of the camera was discussed previously[1]. Version 2.0 is addressed in this paper and represents the current state of the RARE camera development. Version 3.0 is the desirable goal for the camera development to achieve autonomous operation with little or no intervention from a host tracking processor.

	Camera v1.0 TC237 w/New FPGA	Camera v2.0 (New FPGA)	Camera v3.0 (New FPGA)
Interfaces	I/O to PCI	I/O to PCI	I/O to PCI
Max Frame Size	256x256	Full Frame	Full Frame
Exposure Control	Via Frame Rate	User Programmable via Hardware	User Programmable via Hardware
Frame Rate Control	Software Time Base	User Programmable via Hardware	User Programmable via Hardware
Pixel Processor (987) Programmability	Required	Required	Required
Windowing	Software	User Programmable via Hardware	User Programmable via Hardware
DMA	No	Yes	Yes
OS	RTOS	RTOS/Windows	RTOS/Windows
Auto-gain	No	No	Yes
Time-tagging	No	No	Yes
Pixel outputs	One	Two	Two
Multiple Operating Modes	No	Yes	Yes
Large image buffer for data storage	No	No	Yes
Pulsed Time Based Output	No	No	Yes
Sequenced Commands	No	No	Yes
Preprocessing of pixel data	No	No	Yes

Table 1: RARE camera version definitions.

Notes:

1. Time tagging – FPGA maintains time-base information.
2. Multiple operating modes – For streaming video vs single image capture. The configuration is defined by setting bits in control register of the FPGA.
3. Pulsed time base output – FPGA provides a single output which pulses at some prescribed rate.

2 HARDWARE ARCHITETCURE

Release 2.0 of the RARE camera is an expanded version of the release 1.0 design[1]. It is used as part of a real-time target tracking apparatus free-space optical communications and non-invasive eye tracking[2] and provides simplified high-level control of low-level camera operation on an intra-frame basis. The architecture of the release 2.0 RARE camera is shown in Fig. 1 below.

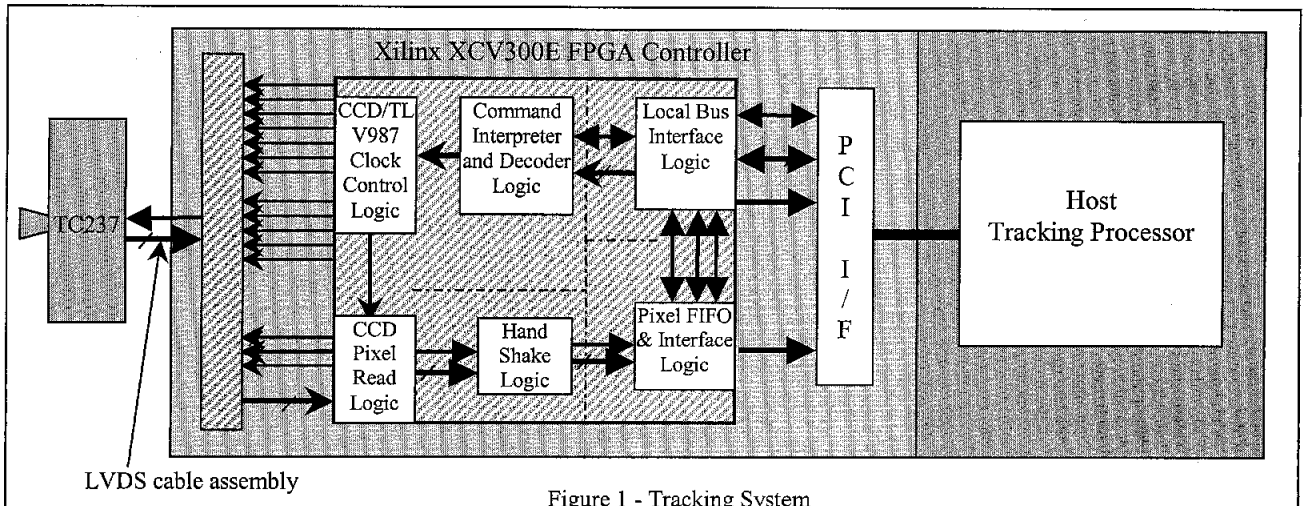


Figure 1 - Tracking System

This figure illustrates the three-component system of the RARE camera consisting of a custom imager card with a low-voltage differential signaling (LVDS) cable assembly, a commercial off-the-shelf (COTS) field-programmable gate array (FPGA) card and a commercial processor (eg personal computer). The imager card was designed with a Texas Instruments TC237 CCD imager chip[3] with two TLV987 signal processor chips[4]. Two 987 processors were required to handle the dual pixel stream output capability of the TC237 CCD. Each processor accepts an analog pixel stream and provides TTL-level output signals to the custom LVDS cable. The cable assembly provides single-ended TTL-level IO signals to the CCD card and the FPGA card but runs differential signals through a pair of SCSI cables to allow for high-speed strobe operation over several feet of cable. The FPGA card is a TransTech PMCFPGA-01 card with a 300K gate Xilinx XCV300E FPGA for the low-level controller of the CCD imager. The host tracking processor is a general-purpose computer with a 32-bit PCI bus used to provide FPGA control parameters and read camera status and pixel data from the FPGA card.

2.1 FPGA registers

The hardware architecture of the RARE camera in Fig 1 utilizes the custom FPGA controller to achieve real-time event driven operation. This philosophy is facilitated by defining registers from a user perspective to include such parameters as ROI size and location, integration time and frame rate to name a few. By customizing FPGA controller operation via these registers, the RARE camera provides low-level CCD imager control based on high-level dynamic situational requirements. Modification of these registers is typically done on a per frame basis, but some registers can also be modified on an intra-frame basis such as multiple ROI read out discussed in section 2.2.

This control scheme is achieved through two types of registers denoted as direct and buried registers. The former are memory-mapped locations that can be written/read directly via a PCI device. The latter are not directly accessible via the PCI but can be configured using RARE camera commands. These commands consist of two, four or six-word sequences, with a word defined as 32 bits. The number of words used in a command depends on the amount of data and address information needed to load a particular buried register or group of buried registers in the FPGA.

2.1.1 FPGA direct register definitions

The directly accessible registers are shown below. A brief explanation of register functionality is given along with the PCI address offset, the register size in bits and write/read capability. These registers typically define the FPGA configuration or initiate an action by the camera.

TEST - 64-bit readable register with PCI offset 0x40H. This register is a dual function location used to read test points within the FPGA or read pixel data from the CCD. The multiplexing logic used to define the mode for this register is setup via the CONTROL register.

CONTROL - 64-bit control and status register with PCI offset 0x48H. This write/read register is used for FPGA configuration control.

PORT0 – 32-bit write only register with PCI offset 0x50H. This register location is used to write two, four or six-word command sequences to configure the buried registers.

CCDREQ – 32-bit write only register with PCI offset 0x70H. A write to this register requests that the CCD controller begin a frame transfer or ROI read out depending on the mode selected.

CCDSTAT – 32-bit read only register with PCI offset 0x74H. The contents of this register provide information pertaining to the state of the controller in either frame transfer or ROI read out modes. In frame transfer mode, this register provides a flag indicating the end of a frame transfer operation. In ROI read out, this register provides access to the counter that tracks the number of available pixels in the pixel FIFO and a flag indicating completion of a ROI read out operation.

CCDSTRST – 32-bit write only register with PCI offset 0x78H. This is used to reset individual bits in the CCDSTAT register.

2.1.2 FPGA buried register definitions

The buried registers are used to define the detailed operation of the CCD control logic. These registers are accessed via the PORT0 (i.e. command port) location and are written to using two, four or six-word command sequences, with a word defined as 32 bits. The buried registers are shown below with a brief explanation of register functionality.

CREG - 32-bit write only register used to define the FPGA control mode. This register currently defines the frame transfer versus the ROI read out mode.

FRAME_REG - 32-bit write only register used to define the dwell time between to frame transfer operations. This register controls the frame rate of the camera.

INT_REG - 32-bit write only register used to define the integration time. The value in this register can be equal to or less than the value in **FRAME_REG**.

SREG - 16-bit write only register used to load the registers in the TLV987 signal processors discussed below.

WIDTH_REGA - 10-bit write only register used to define the width of the ROI. Currently only one set of ROI registers are used. When additional ROIs are folded into the controller, the "A" set of registers will pertain to one of N register sets.

X_ORGA - 10-bit write only register used to define the horizontal starting position (in pixels) of the "A" ROI.

LN_SKIP - 10-bit line skip register used to define how many lines to scroll between line readout operations within the ROI.

HEIGHT_REGA - 10-bit write only register used to define the height of the "A" ROI.

Y_ORGA - 10-bit write only register used to define the vertical starting position (in pixels) of the "A" ROI.

2.1.3 TLV987 video signal processor registers

The TLV987 is an analog signal-processor with an on-board analog-to-digital converter (ADC) used to process one pixel stream of the CCD imager. The signal processor contains a correlated double sampler, 10-bit 27 Mega-sample/second analog to digital converter, programmable gain and dark offset control registers, and a serial interface for register configuration[4]. Software access to the control registers is accomplished via a three-wire interface consisting of the serial data, clock and a chip select line per 987 processor[4].

Release 2.0 allows for setting the TLV987 internal registers to operate in a user-defined mode as dictated by operational circumstances. These registers are loaded as buried registers using six-word commands. The serial register formats and default modes are shown in Table 2 below.

Register	Address	Default Value	Description
Control	0	000 hex	Defines TLV987 configuration
PGA	1	0 dB	Programmable gain; LSB ~ 0.1dB gain
User DAC1, DAC2	2, 3	00 hex	Programmable analog outputs
Course Offset DAC	4	±00 hex	Pre-PGA black and offset correction
Fine Offset DAC	5	±00 hex	Post-PGA black and offset correction
Optical Black Level	6	040 hex	Black level set point
Optical Black Cal.	7	003 hex	Auto black level definition
Test	8	003 hex	Test point definition

Table 2: TLV987 signal processor registers.

2.2 FPGA Implementation Details

Camera operation consists of four fundamental operations. They include the dwell time, the frame transfer operation, line scrolling and ROI read out. The minimum time required to read out a ROI is defined as the sum of the times for the frame transfer, the line scrolling and the ROI readout operations.

Figure 2 shows the sequence of these operations and their associated timing signals. The mode flag is a bit in the control register of the FPGA and its state defines the operation of the camera in response to a request from the host tracking processor. A request to the camera is issued by writing the CCDREQ register and results in pulsing the flag, CCDReq, in Fig. 2. The response of the camera will be a frame transfer or line request depending on the state of the mode flag. If the mode flag is set, a frame transfer will commence as indicated by the frame transfer flag. If the mode flag is low, a combination of pixel line requests and line read out operations will be performed to commence a ROI transfer as denoted by the line valid flag in Fig. 2.

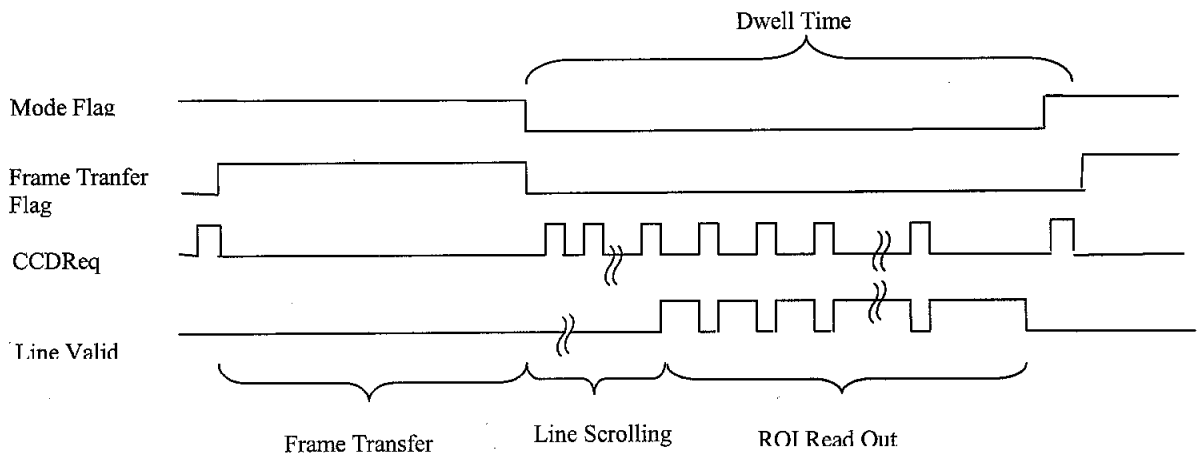


Figure 2 – Camera operations.

2.2.1 User programmable frame rate and integration time control

The frame rate control function uses the FRAME_REG register to store a count value that corresponds to the dwell time and is defined as the number of 20MHz clock cycles between frame transfer operations. This register is used to initialize the dwell time down-counter at the completion of the current frame transfer. The next frame transfer is automatically requested by the FPGA controller when the down-counter goes to zero. During a frame transfer operation, this counter is held at zero until the frame transfer has completed. This functionality provides the host tracking processor with a mechanism to set the integration time and the frame rate of the camera.

A provision is included to allow for independent control of the integration time with a given dwell time. The INT_REG stores the integration time count. This count is compared to the dwell time count and generates a reset pulse to the CCD imaging area when they are equal. This value must be equal to or less than the dwell time count to be meaningful. Values larger than will the dwell count will have no effect on imager operation.

If these registers are set to zero, a frame transfer will occur only when requested by the host tracking processor. This case will result in variable integration time and essentially operates the camera in a single-shot mode. If these registers are set to the same non-zero value, then the camera will request frame transfers autonomously and the dwell time will equal the integration time. In this mode, the integration time can be shortened with respect to the dwell time by setting the integration time register to a smaller value than the dwell time register. In either case, this mode is referred to as the autonomous frame transfer mode.

2.2.2 Single ROI read out

The ROI is defined by the four window configuration registers consisting of the (x,y) origin, X_ORGA, Y_ORGA and the width and height information, WIDTH_REGA and HEIGHT_REGA. Each of these registers is 10 bits wide to allow for ROIs positioned anywhere within the 684 columns and 500 rows of the TC237 field-of-view (FOV) and sized from one pixel to the maximum number of pixels in a dimension. These registers are loaded prior to requesting a ROI read out operation and are used to load counters within the FPGA controller for ROI read out. They are updated only when a new ROI size or location is required.

A request for a ROI read out is initiated by the host tracking processor but proceeds autonomously. The FPGA controller provides pixel FIFO status information from an up/down counter that tracks write/read operations to the pixel FIFO. The count value is accessed by the host tracking processor via the CCD status register, CCDSTAT, and is used to determine the timing and quantity of pixel data available for read out. For large format ROIs, the FPGA controller additionally controls the rate at which pixel data is loaded into the onboard pixel FIFO. This mechanism is used to avoid losing pixel data if the host tracking processor cannot retrieve the pixel data in a timely fashion.

In the autonomous frame transfer mode, the ROI read out time may exceed the programmed dwell time. In this instance, the FPGA controller periodically resets the CCD imaging area, with the time between reset pulses defined by the dwell time. This mechanism allows for variable ROI readout timing due to indeterminate timing on the part of the host

tracking processor but maintains a fixed dwell/integration time. When the ROI read out operation is complete and the FPGA controller is switched back to the frame transfer mode, the camera will begin the next frame transfer after the current dwell period has expired. So ROIs can be read out at whatever rate is required (but not faster than the frame rate) without affecting the integration time.

2.3 Multi-ROI operation

The host tracking processor loads ROI parameters into the FPGA controller to define the ROI size and location. These parameters are used to scroll through unwanted lines and unwanted pixels per line until the ROI is reached. They additionally define the number of pixels per line and the number of lines to read out for the ROI.

Dynamic adjustment of these parameters can be done on a per-frame or intra-frame basis to allow for enhanced system adaptation. They can be used to define a ROI for one or more frames, or they can be adjusted within one frame to allow for multiple ROIs within a single frame. This case is depicted in Figure 3 for two ROIs.

When read out of a ROI(s) is required, the tracking processor sets the mode of the FPGA controller to ROI read out and requests pixel data as defined by the ROI parameters previously loaded. The tracking processor then polls the FPGA controller for available pixel data to initiate the ROI read out from the pixel FIFO in the FPGA. If more than one ROI per frame is required, the tracking processor can load new size and location parameters for the next ROI *without requesting a new frame transfer*. The vertical location parameter of the next ROI is defined relative the last line of the current ROI and denoted by the inter-window scroll area in Fig. 3.

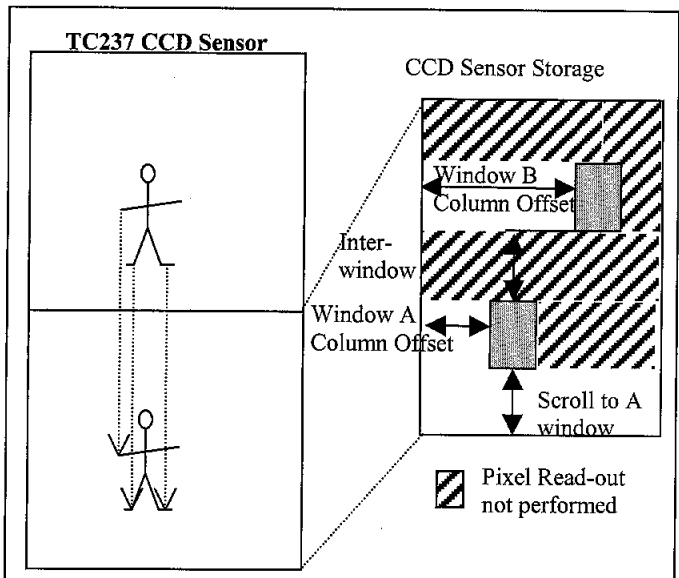


Figure 3 - Frame transfer and ROI read

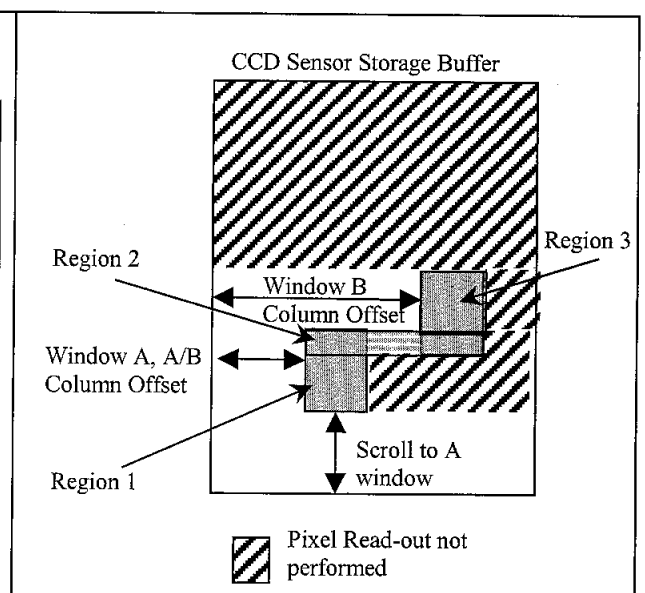


Figure 4 – ROI read out with common rows of pixels.

This methodology also allows for ROIs that overlap or share common rows of pixels with or without vertical separation. The primary difference compared to the case above is in the definition of the ROI. For the case of common lines of pixels between ROIs, the tracking software must read out three different regions corresponding to the two areas where the ROIs do not share common rows of pixels and an additional third region containing pixels from both ROIs. This additional region must read out pixels for both ROIs and will have a width parameter defined by the left edge of the left most ROI and the right edge of the right-most ROI. This methodology is illustrated in Fig. 4 for two ROIs and is applicable to N (>2) ROIs.

3 CAMERA INTERFACE AND SOFTWARE CONTROL

The RARE camera is accessed as a memory-mapped device via the PCI bus of the host tracking processor as shown in Fig. 1. This architecture allows for both programming the Xilinx FPGA and communicating with the FPGA as a CCD imager controller via the PCI bus. The control software running on the host tracking processor communicates to the camera by writing sequences of 32-bit words to the FPGA controller for register initialization and performing various operations. A response from the camera is generated by flags denoting the availability of a new frame within the CCD

storage area or data in the pixel FIFO of the FPGA. This response mechanism provides the synchronization of the host tracking processor software to camera operation without knowledge of detailed camera operation.

3.1 Host Tracking Processor Interface

The COTS FPGA card used for the RARE camera contains a master-slave PCI interface for communication between the host processor and the memory and FPGA on the card. This functionality is achieved with a dedicated PCI interface chip that maps the PCI signals to a local data/address bus structure. The low-level details of the PCI interface are described in the QL5064 user's manual[5].

The PCI interface chip can be programmed for a variety of interface options. As part of the RARE camera, the card is used only as a PCI slave device capable of 32 or 64-bit transfers. Data transfers can be single memory read or write operations, or they can be defined as direct memory access (DMA) transfers for high-speed throughput. The DMA operation can be further defined according to the size of an indivisible block transfer, where the complete DMA transfer consists of one or more block transfers. DMA operation is also specified in terms of running in a simplex or half-duplex mode[6]. These options allow for optimizing DMA operation for a given application.

Within the context of the RARE camera, single accesses are used for loading parameters and reading camera status. ROI read out operations use the simplex DMA mode with a block size of 64, 64-bit words to reduce pixel read out time. The block size comes into to play by allowing for stalling of the DMA operation on block boundaries if not enough pixel data is available for the next block transfer. This mechanism is used as the rate limiting mechanism by the tracking processor to read out data as fast as the CCD imager can supply data without under-flowing the pixel FIFO in the FPGA. This constraint comes from the fact that the CCD card currently has a maximum data rate of 20 mega-pixels per second (for dual pixel stream output) while the PCI DMA operation can sustain a 33 mega-pixel per second rate.

The use of DMA transfers provides for higher throughput over single location accesses by providing dedicated PCI bus operation during a block transfer. The use of the stall mechanism additionally allows for simultaneous reading of pixels from the pixel FIFO in the FPGA while pixels are being written into the FIFO from the CCD storage area. This methodology effectively limits the camera throughput to the CCD imager readout rate with a small latency to load the first block of pixel data into the FIFO.

3.2 Software Flow

A flow chart of the host tracking processor software is shown in Fig 5. The basic functions of the tracking processor are to provide control information to the camera and read and process pixel data from the camera. The control is in the form of initialization and command sequences used to operate the camera. The results from the processed data are for target recognition and tracking via adjustment of the window size and location parameters.

The flow begins by initializing the TLV987 signal processor chips and setting the FPGA controller for single or dual pixel stream operation with the mode flag set for frame transfers. A request for a frame transfer is issued followed by initialization of the DMA process used to read out the first ROI. After completion of DMA initialization, the CCDSTAT register is polled to determine when the frame transfer is complete. Upon sensing this event, the mode is switched to ROI read out and the DMA operation commences. After completion of the DMA operation, the controller mode is switched back to frame transfer operation and the retrieved pixel data is displayed and/or processed. If the autonomous frame transfer mode is being used, the DMA process is immediately initialized in preparation for the next ROI read out. If the manual frame transfer mode is being used, a new frame transfer request is issued before initializing the next DMA operation. This sequence of frame transfer followed by ROI read out is maintained for the duration of a tracking experiment.

3.3 Software Control

The tracking processor software controls the RARE camera by sending commands to the FPGA card. They are a combination of writing to the direct or buried registers in the FPGA followed by accessing the CCDREQ register. The format for writing into the buried registers consists of a six 32-bit words: the first word is null; the second word defines the command; the third word is null; the fourth word contains the address of the register or set of registers; and the fifth and six words contain the data for the register(s).

The host tracking processor uses a variety of register write operations to define higher-level commands. They provide the primary means of camera control and synchronization. The various commands defined are shown below.

InitializeFPGA –loads the registers in the FPGA controller and TLV987 signal processors with default values.
InitializeGainControl –sets the gain registers in the 987 signal processors
DeletePF - Releases the FPGA device.
FirstFrameTransfer –initiates the frame transfer operation by setting the FPGA controller in the frame transfer mode and requesting the first frame transfer operation.
SubsequentFrameTransfer – sets the FPGA controller to the frame transfer mode to resume frame transfer operations.
InitializeSingleLineRead – set the line scroll register to load one line of pixels into a shift register for single line read out.
InitializeDualLineRead – set the line scroll register to load two lines of pixels into the shift registers for dual line read out.
FetchImageData – get an image from the CCD.
GetFIFOPixel – get a single pixel from the pixel FIFO.
GetDMAPixel – get pixels from the pixel FIFO using DMA read out with one pixel per 64-bit FIFO location.
GetDualPixels – get pixels from the FIFOusing DMA read out with two pixel per 64-bit FIFO location.
GetFirstLinePixel – read out the odd field of pixel data.
GetSecondLinePixel – read out the even field of pixel data.
SplitValue - Divide a 10 bits value into two parts; the upper part contains the 8th and 9th bits; the lower part contains bit-0 to bit-7.
SetCCDControl - Set a TLV987 register to a specific value. The TLV987 register can be the control register, the CoarseDAC register, or the CCDGain register.

Frame rate/Exposure Control – defines the amount of time the CCD spends collecting charge for a particular image and the rate at which frame transfer operations occur. Exposure control allows for clearing of the imaging area without performing a frame transfer.

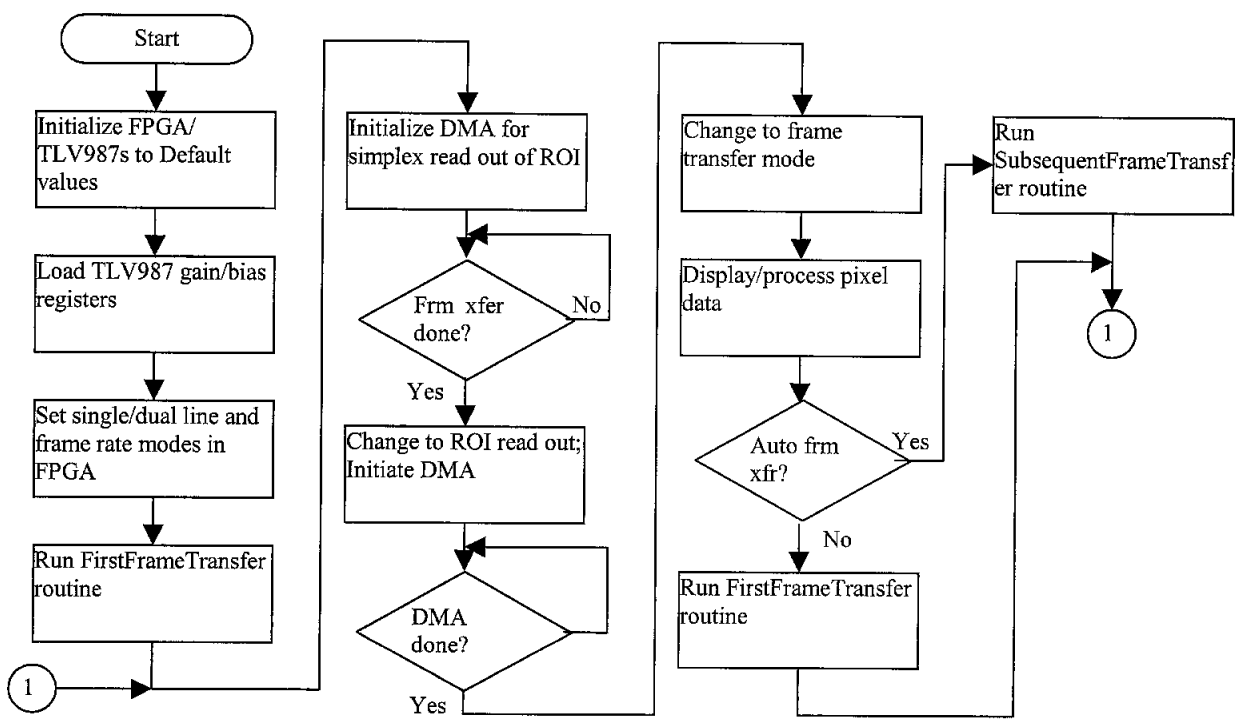


Figure 5 – Tracking processor control software flow chart.

4 EXPERIMENTAL RESULTS

A series of benchmark experiments were performed on the RARE camera to characterize the actual frame rate for a single ROI per frame. The results show the achievable frame rates for different sizes and locations of ROIs within a frame. These results are compared to the maximum theoretical frame rates, which include the CCD frame transfer operation, ROI size and location definition, data transfer from the CCD to the pixel FIFO in the FPGA and transfer of pixel data to DMA memory in the host tracking processor.

4.1 Experimental setup

The setup for the benchmark experiments used an IBM compatible PC with an AMD 2000+ CPU and 512M DRAM for the host tracking processor. The operating system is Windows 2000, SP2. The FPGA card driver is version pf_1.3r0 from Transtech and is commercially available.

The customized firmware in the FPGA for local camera control is version pf1t_81v1b and supports the functionality in Table 1. The TC237 CCD is operated with dual pixel stream outputs with each stream running at 10 mega pixels per second. Dual pixel stream read out refers to transfer of pixel data from the CCD storage area to the FPGA pixel FIFO. Read out of pixel data from the pixel FIFO to the host tracking processor memory uses DMA accesses. The experimental results do not account for time needed to process or display the pixel data.

The autonomous frame transfer mode is disabled in the experimental trials so that frame transfer operations occur only when requested by the host tracking processor. Thus, the dwell and integration time are governed by the ROI read out time. The frame transfer time for firmware version pf1t_81v1b is 100 micro seconds.

4.2 Experimental results

The experimental and theoretical results are presented in Figures 6 and 7. The measured results are presented by the solid curves and the theoretical results by the dashed curves as indicated by the legends at the right of these figures. The horizontal axis represents the ROI size or location in pixels and the vertical axis represents the rate at which images are received from the CCD. For both experiments, the ROI has an NxN aspect ratio.

Figure 6 shows a family of curves where each curve represents a particular origin position for the ROI. The origin corresponds to the upper left corner of the ROI and is the location of the first pixel read from the CCD. The curves in this figure correspond to ROI origins located at pixel positions (0,0), (100,100), (200,200), (300,300) and (400,400). The shape of a curve shows the change in frame rate as a function of the ROI size for a given origin position.

The uppermost curve in Figure 6 represents the theoretical frame rates for ROIs with origin located at pixel 0,0. The theoretical rate is calculated by adding together the time for all events needed to perform frame transfer and ROI read out. For a 10x10 ROI, the theoretical frame rate is approximately 8.5KHz. For a 100x100 ROI with origin at (0,0) the theoretical rate is 1.4KHz. The corresponding experimental numbers are 6.4KHz and 1.1KHz.

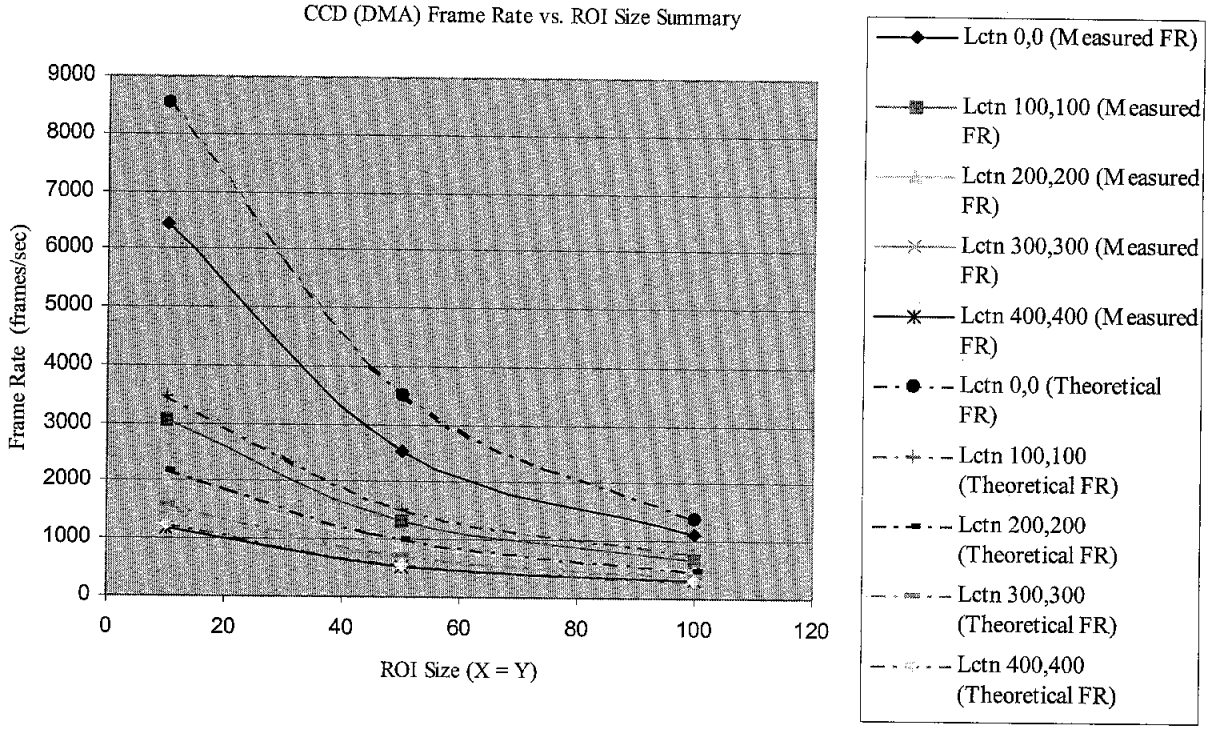
Figure 7 shows a family of curves where each curve represents a particular NxN ROI size. The shape of each curve shows the change in the frame rate as a function of the ROI origin position. The ROI origin is allowed to move along a diagonal line such that the horizontal and vertical displacements of the origin are equal. The curves shown in this figure are for ROIs of size 10x10, 50x50 and 100x100.

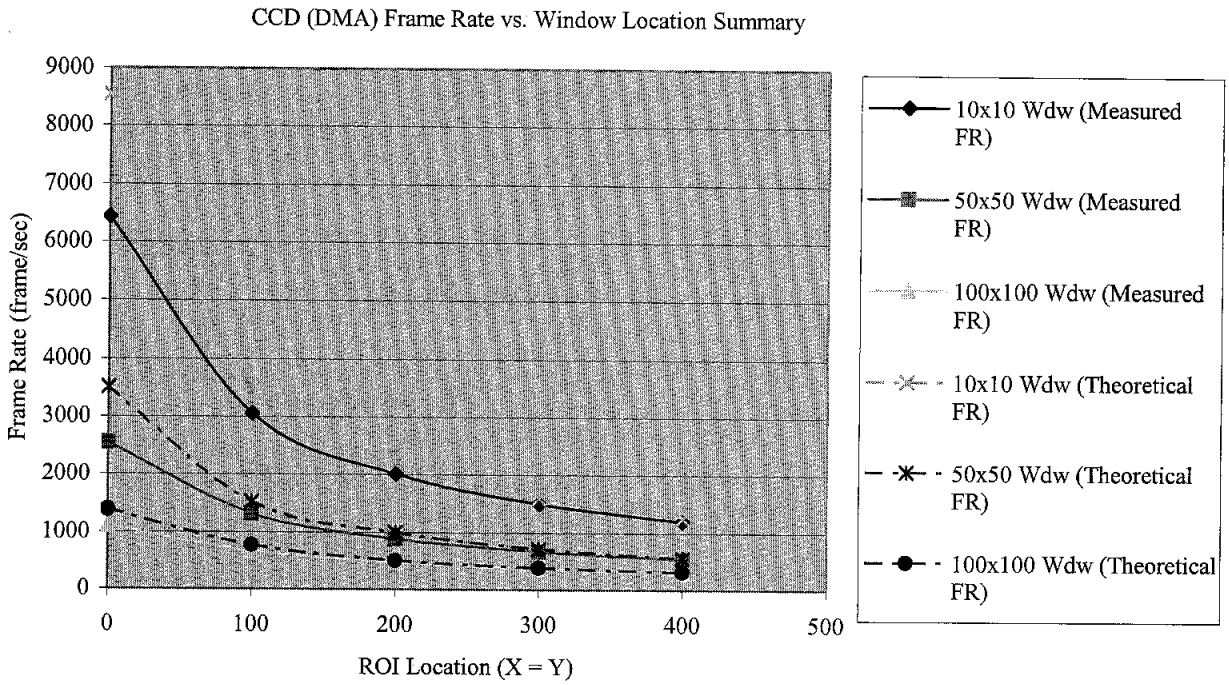
The uppermost curve in Figure 7 shows the theoretical frame rates for 10x10 ROIs. The corresponding experimental curve is the uppermost solid line. For the origin position of (0,0) and (400,400), the theoretical rates are 8.5KHz and 1.2KHz respectively. The corresponding experimental numbers are 6.4KHz and 1.1KHz.

The experimental frame rates are approximately 75% of the theoretical rates. The difference between the theoretical and measured frame rates is due to the following factors. A 20 micro second delay was measured between the end of writing the ROI data into the FPGA pixel FIFO and the next frame transfer request. This delay is due to the time needed to complete the DMA read out from the pixel FIFO and switch to the frame transfer mode and request the next frame transfer operation.

The predominant portion of this 20 micro second delay is due to the DMA completion time. This time will always be present due to the need to rate limit the DMA read out as a means of preventing FIFO underflow. The FPGA control logic stalls the DMA operation when the number of pixels in the FIFO drops below a pre-defined threshold. This condition persists until the ROI data has finished loading into the pixel FIFO. Once completed, the DMA is allowed to read all data from the FIFO. The 20 micro second delay can be reduced by dropping the threshold but this amount of

time is small compared to the 100 micro second frame transfer time. Its effect is most notable when the ROI origin is located at (0,0).





There is an additional variable time delay due to a discrepancy in the theoretical line request timing as compared to actual hardware operation. This delay is proportional to the number of lines that need to be scrolled before reaching the first line of the ROI and the number of lines requested during ROI read out. The effect of this delay will be most notable for ROIs located in the lower right quadrant of the CCD FOV.

5 CONCLUSIONS

The goal of the RARE camera development is to provide a key component for a real-time, adaptive tracking platform. We have developed this infrastructure by implementing a methodology to quickly extract pertinent pixel data using a commercially available progressive or interlace scan imager. This methodology is also well suited to adjusting the camera parameters to accommodate changing ambient and target conditions during tracking.

In this paper, we presented details of the RARE camera design based on the Texas Instruments TC237 CCD imager chip. The novel feature of this design is the use of an event-driven paradigm for imager control. This capability was implemented by developing a custom FPGA controller that can convert a commercially available CCD imager into a smart pixel device.

Communication to the FPGA controller is via commands from a host tracking processor. This combination of FPGA controller and host tracking processor provides for higher-level commands to handle low-level imager operation for dynamically controlled ROI capability on a per-frame and intra-frame basis. The result is the version 2.0 RARE camera design with features listed in Table 1. Additional features are envisioned for the version 3.0 design as shown in Table 1.

To assess the speed performance of the version 2.0 design, two experiments were conducted. The first experiment illustrated the slow down in frame rate for a fixed ROI origin but increasing ROI size. The second experiment illustrated the slow down in frame rate as a constant-sized ROI is moved away from the first row (top) and first column (left) of the CCD FOV. The maximum achieved frame rate was 6.4KHz for a 10x10 ROI with origin located at (0,0).

The results from these experiments achieved 75% of the theoretical best possible frame rates for this CCD imager in the configuration detailed in section 4.1. The system slow-down with ROI position or size is an inherent characteristic for a progressive scan or interlace imager. A nearly two times speed up is possible by running the imager at its maximum possible clock speed of 20 MHz per pixel stream. Additional speed increases require using a different progressive scan or interlace imager with more than two pixel outputs to provide more parallelism in accessing the image data. Due to the generality of the RARE camera control scheme, the FPGA controller can be used for other commercially available CCD imagers to optimize system performance in terms of speed, image quality or other parameters of interest.

ACKNOWLEDGEMENTS

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

1. Steve P. Monacos, Angel A. Portillo, William Liu, James W. Alexander, Gerardo G. Ortiz, "A High Frame Rate CCD Camera with Region-of-Interest Capability", *2001 IEEE Aerospace Conference Proceedings*, Big Sky, Montana, March 10-17, 2001.
2. Ashit Talukder, John-Michael Morookian, Steve Monacos, Raymond Lam, Clayton Labaw , "Real-time Non-Intrusive Eyetracking and Gaze-point Determination for Human-Computer Interaction and Biomedicine", *2nd WSEAS International Conference on Signal, Speech and Image Processing* (WSEAS ICOSSIP 2002), September 25-28, 2002.
3. Texas Instruments, *680x500 Pixel CCD Image Sensor TC237*, June 1996.
4. Texas Instruments, *3-V 10-Bit 27 MSPS Area CCD Sensor Signal Processor TLV987*, September 1999.
5. QuickLogic Corporation, *QL5064 User's Manual, Revision A*, December 1999.
6. Transtech Corporation, *PMC-FPGA01 User Manual*, 2001.