A High Performance Micro Channel Interface for Real-Time Industrial Image Processing Applications

Thomas H. Drayer, Joseph G. Tront, and Richard W. Conners

Bradley Department of Electrical Engineering, Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061-0111 USA; phone: (703)231-6646; e-mail: tdrayer@birch.ee.vt.edu, jgtront@vtvm1.cc.vt.edu

ABSTRACT - Data collection and transfer devices are critical to the performance of any machine vision system. The interface described in this paper collects image data from a color line scan camera and transfers the data obtained into the system memory of a Micro Channel-based host computer. A maximum data transfer rate of 20 Mbytes/see can be achieved using the DMA capabilities of the camera interface. Programmability of the interface provides flexibility in selection of features such as collected field-of-view, data format, collection method, and camera control. System design is such that the speed of the interface allows for the collection of images intended for real-time processing by any available Micro Channel resident processor(s).

INTRODUCTION

The camera interface described here is part of a machine vision system that is primarily used to apply known web inspection techniques to the analysis of rough-cut lumber. The same techniques used to collect and process images of rough-cut lumber can very easily be extended to a number of other industrial inspection applications, including the inspection of textiles, copper tubing, and rolled steel.

In this system, image data is collected from camera sensors and transferred by the camera interface through the Micro Channel bus into available system memory. Machine vision algorithms are used by the system processor(s) to detect defects such as knots, wane, cracks, splits, and stains. A *critical defect* is one whose effect on the final product is significant enough to warrant its removal. Knowledge of the location of critical defects allows the system processor(s) to develop an optimal cutting scheme that reaps the highest yield from the input lumber.

Performance requirements of the camera interface are derived from rough cut lumber processing requirements. The speed at which input lumber is processed by the vision system, i.e., the *system throughput,* is an immediate parameter of concern. Current manual technology is capable of a sustained system throughput of approximately two linear-board-feet/sec. The objective of this system is to scan boards and perform the necessary machine vision algorithms to decide on an optimal cutting strategy for the input lumber. The algorithms should be optimized such that:

- the output lumber is free of defects,
- the volume of clear wood output is maximized,
- and the system performs as fast as its manual counterpart.

An automated system to inspect rough cut lumber has many advantages over the current manual technology. These include the improved reliability, accuracy, and consistency associated with automated systems.

SPECIFICATIONS

To process the majority of types of lumber used by the furniture industry, a machine vision system must accept boards of sizes up to 12 inches wide by 16 feet long. The required image resolutions have been experimentally determined by examining and processing board images at varied spatial resolutions. It has been determined that accurate image perception is achieved at cross-board resolutions of 64 points/inch and down-board resolutions of only 32 points/inch [1]. These two pieces of information determine the volume of data that will be collected. When both sides of a 16 foot board are seamed at the above resolutions, over 28 Mbytes of image information must be transferred to the computer's memory and processed.

Data throughput is defined as the amount of data per unit time that is continuously transferred from the linescan camera to system memory. When data is collected from both sides of a board, these process speeds and image resolutions require the machine vision system to collect and process 3.5 Mbytes of image data per second.

High data rates are characteristic of most industrial inspection applications. This machine vision system may require even higher data rates if X-ray, CT, or range information is to be collected. To allow for future system expansion to include data collection from these additional sources, system resources such as bus bandwidth are conserved by this interface design.

The above requirements are the standards for optical wood processing systems proposed by Conners in [1]. A *real-time* system is defined as any system which collects and processes board images at the above resolution and speeds. A system that can operate within these guidelines is a reasonable candidate to replace the current manual technology.

SYSTEM COMPONENT DESCRIPTION

The overall machine vision system consists of two important subsystems: the materials handling subsystem and the image processing subsystem. Materials handling components are used to transport an object, in this case a rough-cut board, through the system at known speeds and positions. The image processing

0-7803-1328-3/94$03.00

subsystem is used to collect and process spatial image data obtained from the object currently in the system. Two essential image processing elements are interfaced: a camera sensor and associated control unit and a Micro Channel-based host processing unit. The actual components used in the developed machine vision system are a Pulnix TL2600 RGB line scan camera and an IBM PS/2 Model 90. The camera interface allows these two image processing elements to communicate. A basic knowledge of these two system elements is required to understand how they are interfaced.

A *linear array camera* (line scan camera) consists of a single one-dimensional array of pixel elements. The full array of element intensities is called a frame (scan) of data. Color image frames are comprised of three one-dimensional arrays; called the red, green, and blue channels. The intensity of red, green, and blue light at each pixel location is determined by the camera sensor, then quantized to a digital representation by the camera control unit. The pixel values are quantized to 3 eight-bit color values, yielding a *full-color* image. Eight-bit digital values are sequentially presented by the camera control unit to the camera interface using the synchronous protocol defined in [2].

A linear array camera is preferred for this and many other industrial inspection applications, since it allows images of varying sizes and spatial resolutions to be easily collected. For example, the different cross- and down-board resolutions are easily obtained by controlling the integration time and object velocity using a linescan camera, without the use of special optical hardware.

The Micro Channel system bus is used in the host processing unit because of its high bandwidth, data path width, large memory address space, and support of multiple bus masters. This high speed bus architecture was developed by IBM for the new generation of 32-bit processors, such as the Intel 80386 and 80486. The multitasking, multi-user Micro Channel bus is designed to replace the single-task, single-thread Industry Standard Architecture (ISA) bus used in IBM PC/AT systems. The minimal implementations of the Micro Channel bus allow instantaneous transfer rates of up to 20 Mbytes/see [3, 4]. Future upgrades will provide instantaneous transfer rates of up to 160 Mbytes/see [4, 5]. A 32-bit address space is supported, allowing up to 4 Gbytes of installed system memory.

A significant enhancement of the Micro Channel over previous IBM bus architecture is the ability to support multiple bus masters. Additional processors or DMA devices can transfer data across the channel after being granted bus mastership. This allows multiple bus masters to coexist on the single system bus.

INTERFACE DESIGN

Design of the camera interface has a distinct goal, i.e., to make camera image data available to Micro Channel resident system processors in a fast, efficient, and flexible manner. This is accomplished by sequentially transferring image data bytes from the camera interface, across the Micro Channel bus, and into system memory. To this end, the method of enacting these data transfers is a critical design choice.

Three methods are available to transfer image data into Micro Channel system memory. First, a remote device such as an I/O processor or the system processor can read in data from the interface board and then transfer it out the appropriate location in Micro Channel system memory. Image collection devices such as 4MEGVIDEO Model 12 from EPIX Inc. use this method of transfer on the ISA bus [6]. Since two Micro Channel bus cycles are rewired for each transfer, bus bandwidth is not optimally conserved in this transfer method. As a second design alternative, the image data can be transferred by the camera control unit into a large amount of Micro Channel system memory that is physically located on the interface board. This design alternative is used by most commercially available line scan interface boards for the Micro Channel, VME, or ISA busses [7, 8]. One disadvantage of this alternative is that it requires a very large memory subsystem to reside on the interface board, outside of the main computer system. A second disadvantage is that some new computer architecture use the bus for I/O devices only, and the system memory is located on a separate, high bandwidth bus structure which is tightly coupled to the processor. This is the case for Micro Channel-based IBM RS/6000 POWERstations, in which a system memory is located on a separate bus [9].

Finally, the camera interface can be designed as a Micro Channel bus master. In this design alternative the image data is transferred into system memory independently of system processor(s) using Direct Memory Access (DMA). To enable the transfer using DMA, the camera interface obtains control of the Micro Channel bus and drives all the necessary bus signals required to transfer the image data independently. A line scan camera interface utilizing DMA on the EISA bus is currently available from MATROX Inc. [10], but the authors are not aware of any line scan camera interfaces that are available for the Micro Channel. This DMA bus master approach is the best choice for data collection in image processing and other high data throughput applications. It uses half of the bus bandwidth of the first design alternative, and unlike the second design alternative it allows the memory subsystem to be independent of the camera interface.

BUS MASTER DESIGN

Any Micro Channel bus master must have certain functional capabilities in order to operate properly. Illustrated in Figure 1 are six logical subsystems required for any Micro Channel DMA bus master camera interface. These subsystems work together to provide an efficient data collection and transfer device.

The **POS register** subsystem is required by convention for all cards that are inserted into Micro Channel adapter slots. Static system attributes such as bounds on the card's I/O and memory address spaces, arbitration level, device ID, and interrupts used are all programmed into the POS registers by system level routines during power-up. The values are established by a setup program that is run after any Micro Channel cards are inserted or removed. This same program also stores the values in non-volatile memory. This information is stored in up to eight l-byte locations that are formatted according to the guidelines given by IBM in [3].

The **I/O port** subsystem allows the camera interface and the system processor to communicate. More specifically, the system processor programs the options of the camera interface by writing bytes of information to programmable registers on the camera interface. This is accomplished with a Micro Channel I/O write cycle. Options such as the camera’s field-of-view, desired color channels, data transfer destination, and data format are programmed in this manner. I/O read cycles are used by the system processor to monitor the status of the camera interface by reading bytes from other I/O port locations on the camera interface. Using this facility, the interrupt and job completion status can be polled.

Port locations are implemented with standard 8-bit latches or registers with address decode logic included to determine

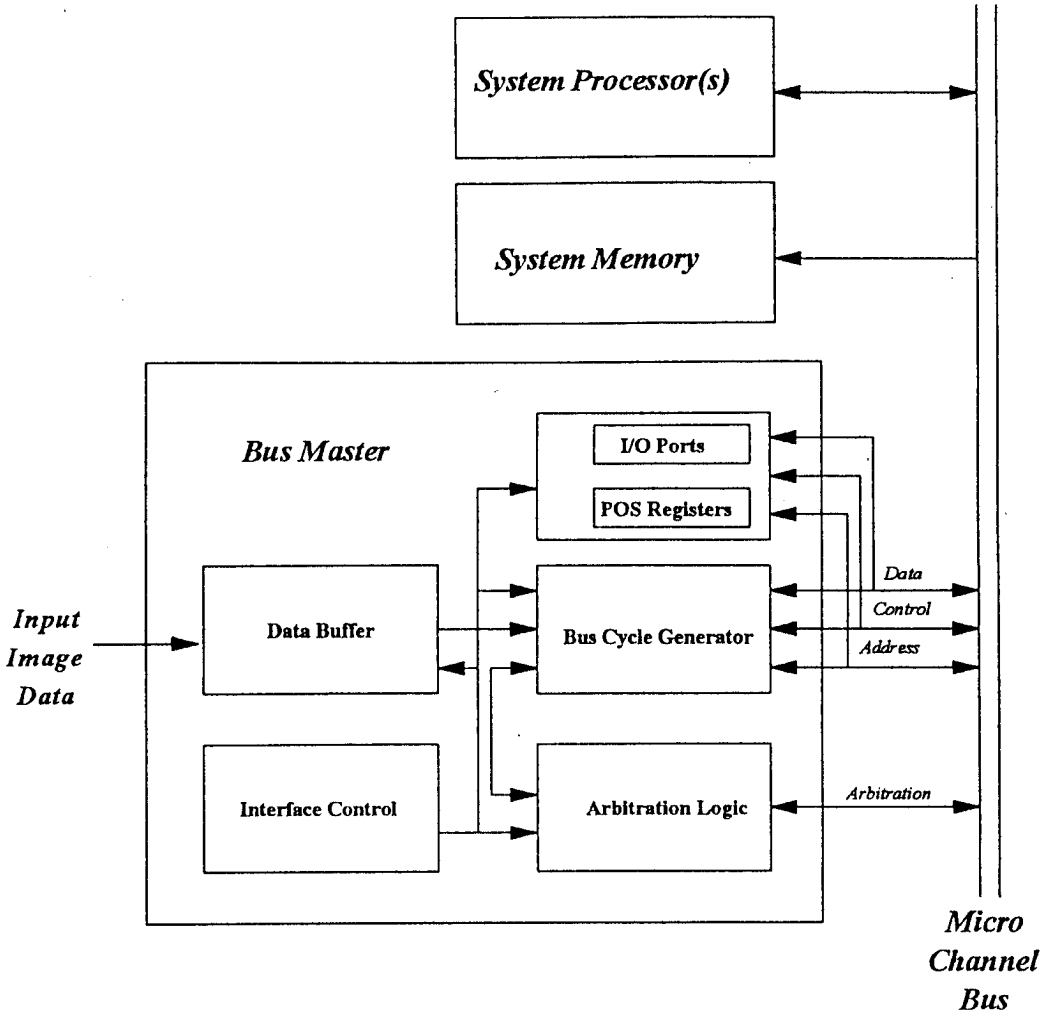

Figure 1. Micro Channel bus master system diagram

when the camera interface's port locations are accessed. Information stored in the POS registers establishes the I/O space of the card. This information is used along with the timing signals obtained from Micro Channel control signals to fully decode I/O bus cycles.

The **arbitration logic** subsystem is used to obtain ownership of the Micro Channel bus. Although multiple bus masters may exist on the bus concurrently, only one may own the bus at any given time. This prevents more than one device from driving the signal lines at the same time. One of sixteen unique arbitration levels is stored in a POS register of each Micro Channel bus master. This unique level is used during special arbitration bus cycles to identify and prioritize the bus masters which desire ownership of the bus. The arbitration bus cycle awards bus ownership to the highest priority bus master participating in the cycle. After the camera interface obtains bus ownership, image data transfer can begin.

After receiving bus ownership, either a single transfer or a number of transfers can be made before bus ownership is relinquished. If a single transfer is made the arbitration logic must immediately relinquish bus ownership after the transfer is complete. When several transfers are made, two methods of transfer exist. First, the normal default bus cycle, in which the address is presented followed by the data, is used for each transfer. This transfer method is called the **burst mode.** The second type of transfer can only be used when all transfers are made to consecutive memory locations. In this transfer mode, the address for the first transfer is presented followed by the data as before. For the following transfers, only the data is required, as the address is implied. This transfer method is called the **streaming mode,** and uses half the bandwidth of the burst mode. Streaming mode is not supported by all implementations of the Micro Charnel. For example, the Micro Channel implementation of the IBM PS/2 Model 80 [11] does not support the streaming mode.

All bus masters must relinquish the Micro Channel bus within 7 microseconds from the time when another bus master requests bus ownership [3]. For either the burst or streaming modes, two methods of bus ownership termination are possible. First, a programmed number of transfers that require less than 7.5 microseconds can be made. Second, the arbitration logic may

recognize when another bus master desires use of the bus, and relinquish ownership within 7.5 microseconds. Since the second method generally allows more transfers per bus ownership, the overhead associated with obtaining the bus is reduced, because fewer bus arbitration cycles are required to transfer the same amount of information. Therefore, the second method of termination represents the most efficient use of available bus bandwidth.

The **bus cycle generator** transfers data into system memory using Micro Channel bus cycles. Image data and address information for the transfer are obtained from the data buffer and I/O port subsystems, respectively. The bus cycle generator combines the address and data information and supplies timing and control signals to generate valid Micro Channel bus transfers.

The transfer of data into Micro Channel system memory is an asynchronous event, while the data is presented synchronously by the camera control unit. Therefore, the image data must be stored in a **data buffer** in the camera interface until it can be transferred to its destination. Since data is collected as 8-bit bytes and transferred in 32-bit Micro Channel transfers, some data formatting within the camera interface is required. The implementation of the data formatting hardware may be significant if a number of programmable data format options are desired.

The size of the buffer should be sufficiently large such that the buffer never completely fills with data. If a byte of image data arrives when the buffer is full, a buffer overflow condition occurs and the data is lost. The *maximum bus ownership latency* is defined as the maximum amount of time required to obtain ownership of the Micro Channel bus. Information on the number and type of installed Micro Channel bus masters can be used to determine this maximum latency as illustrated in [3]. Since the image data input rate is known, the buffer storage requirements can be calculated from the maximum bus ownership latency. If full color image is presented by the camera control unit synchronously at a rate of 3.5 Mbytes/see, the minimum buffer size is 473 Bytes.

The functions of the camera interface are enabled and disabled by the **interface control** subsystem. Control signals are generated here to coordinate the operation of all of the separate interface subsystems.

The logic required to implement all of these subsystems can be significant if a substantial amount of programmability is to be provided. Micro Channel chip sets are available from a number of manufactures that incorporate several of these subsystems on a single chip [12, 13, 14]. These high-integration chips allow very flexible systems to be designed and built with a minimum of design and debug effort.

PHYSICAL CONSTRUCTION

Two fully operational prototype camera interfaces have been designed and constructed using the preceding design principles. A third prototype is currently under development. The current prototype is wire-wrapped and fits into any compatible 32-bit Micro Channel slot. A printed circuit board version of the camera interface is also currently under development. Utilizing the six previously defined subsystems, a brief description of the implemented camera interface's capabilities are provided in the following paragraphs.

Only four of the eight possible POS byte locations are implemented. These locations define the interface I/O address space, arbitration level, card identification number, and interrupt level of the camera interface. A Chips & Technologies chip 82C612 [12, 13] is used for both bus arbitration and for addressing the discrete TTL latches used to construct the POS registers. This standard part was developed for Micro Channel interfacing.

The I/O space contains 32 byte locations. Write-only ports are used to store the destination address of image data and to select data format options. Read-only ports are used to read buffer overflow, interrupt pending, and job completion status. The I/O address decode logic is constructed from TTL comparator chips, using the information stored in POS registers to compare with the current bus address. Both input and output ports are constructed from discrete TTL latches or are incorporated into components such as address counters..

The bus arbitration logic is constructed to provide the most efficient use of Micro Channel bandwidth. An Intel 82C54 programmable timer [15] is used to extend bus ownership near the maximum time of 7.5 microseconds for best utilization of bus bandwidth. The arbitration logic will not attempt to regain control of the bus until all other bus masters have obtained bus ownership, following the fairness algorithm outlined in [3] .

There are three types of valid bus cycles available for data transfer using the Micro Channel, defined as the basic transfer cycles in [3]. When the 32-bit, 200 nsec default cycle is used, an instantaneous transfer rate of 20 Mbytes per second can be achieved. Micro Channel memory that is not fast enough to support the default cycle, must extend the cycle using either the synchronous or asynchronous extended cycles. The bus cycle generator is capable of creating control signals to generate any of these three cycles. A special purpose state machine was developed and implemented using two 8-cell Intel 5C031 EPLDs [16] to generate the correct timing for various bus cycles. When the bus cycle generator is operated at a clock frequency of 20 MHz, an instantaneous data throughput of 20 Mbytes/see can be achieved.

The data buffer is implemented using four separate IDT lKx8 FIFO SRAM memory chips [16] with an access time of 70 nanoseconds. This 4 Kbyte buffer far exceeds the 340 byte minimum buffer size calculated above. The four separate FIFO memories are used to simultaneously provide the 4 bytes of image data required for 32-bit Micro Channel transfers.

To allow efficient processing by the system processor(s), the data buffer allows programmability in the formatting of the transferred image data. Several format options are available. First, each of the red, green, and blue color channels may be individually enabled. Another option allows the field-of-view to be limited to a programmable window within each full scan. Therefore, image data from background pixels and non-desired color channels need not be collected or transferred.

One final option allows the data to be sent to one or several continuous blocks of system memory. The three color channels can be written to consecutive locations of a single block of memory, in the order that they were received from the camera controller. Alternatively, the color charnels are separated into the three different color channels as they are received. A different block of' memory is allocated for each color channel, and the individual color channels are written to consecutive locations of each block of system memory in the order that they were received from the camera controller. This second method allows the individual color channels to be accessed as separate sub-images, which may simplify the indexing required by machine vision algorithms.

The interface control logic coordinates the collection and transfer of image data. Image data is collected when a control bit is set in the I/O space of the camera interface. The data transfers are only initiated when the buffer is half-full. Transfers remain

enabled through several cycles of obtaining and releasing bus ownership, until the buffer is empty. A second simple state machine is implemented in a single Intel 5C031 EPLD to perform this function.

The fully assembled camera interface has been installed and tested in an IBM PS/2 Model 90, interfacing with a Pulnix TL2600 RGB Color Line scan Camera. An image of a rough-cut oak board collected with this system at the required resolutions and speeds is presented in Figure 2 (This image was originally fuIl color, although it is reproduced in greyscale for this publication).

The measured performance of the installed interface has been found to be significantly less than the than the maximum throughput of 20 Mbytes/sec. The average time required for 30 test transfers, each consisting of 1.8 Kbytes of image data, is measured to be 181.3 microseconds. This represents a continuous throughput of 9.94 Mbytes/sec. Two factors contribute to this. First, the installed memory is not fast enough to run the 200 nsec default cycle. Individual transfers were observed to require a 300 nsec asynchronous extended cycle, providing an instantaneous throughput of 13.33 Mbytes/sec. The sustained throughput is also reduced by bus overhead associated with memory refresh and bus arbitration cycles. Activity by other Micro Channel devices, such as the installed SCSI controller and XGA graphics card could reduce the throughput below observed level. Even so, the obtained data transfer rate of 9.94 Mbytes/see provides adequate throughput for most real-time machine vision applications, significantly better than the 3.5 Mbyes/sec required for real-time collection of images at required speeds and resolutions for the rough-cut lumber inspection problem.

A final prototype is currently under construction. Two Xilinx FPGA chips (an XC 4005A-5 and an XC 4008-5) add functionality and increase logic density [18]. The final design will allow collection from 8 independent color channels . This allows a single Micro Channel interface card to collect from two color cameras simultaneously.

CONCLUSION

This paper describes the design of an efficient and high throughput data collection and transfer device for Micro Channel based machine vision systems. A prototype has been constructed and an image is presented as proof of the functionality of the camera interface. When installed in a Micro Channel machine, the interface has successfully transferred image data at a sustained throughput of 9.94 MBytes/see and an instantaneous transfer rate of 13.33 Mbytes/sec. This data rate is sufficient for the real-time collection of images of rough-cut lumber. A maximum instantaneous throughput of 20 MBytes/see could theoretical y be achieved by installing faster Micro Channel system memory. This design provides a high performance component for a real-time industrial inspection/machine vision system.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the Southeastern Forest Experiment Station of the US Forest Service, in particular Philip Araman, for funding this project. Thanks also to Bob Lineberry for his technical assistance during the construction of the hardware prototypes.

Figure2. Collected image of an oak board.

REFERENCES

[1] R.W. Conners, C.T. Ng, T.H. Drayer, J.G. Tront, D.E.Kline, and C.J. Gatchell, "Computer Vision Hardware System for Automating Rough Mills of Furniture Plants," in *Proceedings of SPIE, Applications* of *Artificial Intelligence VIII,* April 17-19, Orlando, Florida, 1990, pp. 777-787.

[2] Pulnix America Corporation, *TL-2600 RGB Colour Linescan Camera Operating Instructions,* Pulnix Inc, pp 1-25.

[3] International Business Machines Corporation, *IBM Personal System/2 Seminar Proceedings,* Volume 5, Number 3, May 1987, pp. 1-70.

[4] International Business Machines Corporation, "Overview of Extended Micro Channel Functions," *Personal Systems,* Issue 4, 1989, pp. 53-60.

[5] International Business Machines Corporation, "New Micro Channel Features," *Personal Systems,* Issue 4, 1989, pp. 61-65.

[6] EPIX Inc., *4MEGVIDEO Model 12,* Sales Literature, 1993.

[7] Data Translation, *Image Processing Handbook,* 1993.

[8] Vision Modules, Inc., *3031 Line-Scan Camera Interface Board,* Sales Literature, 1993.

[9] IBM Inc, *POWERstation and POWERserver Hardware Technical Infoormation,* 1992.

[10] Matrox Inc., *Matrox Magic,* Sales Literature, 1993.

[11] IBM Inc. "IBM PS/2 Model 80 Hardware Technical Reference", 1989.

[12] J. Figueroa, "Implementing Programmable Option Select on IBM's PS/2," *Microprocessor Report,* Sept. 1987.

[13] Chips and Technologies Corporation, *82C611, 82C612 Microchips Micro Channel Interface Parts: Preliminary,* 1987.

[14] Micro Design Resources, "C&T Microchips Simplify Micro Channel Interfacing," *Microprocessor Report,* Sept. 1987.

[15] Intel Corp. *Microprocessor and Peripheral Handbook Volume II Peripheral,* 1988.

[16] Intel Corp., *5C031 Programmable Logic Device,* 1987.

[17] IDT Inc., *Specialized Memories,* 1993.

[18] Xilinx Inc., *The XC4000 Data Book,* Xilinx Inc, 1993.

PROCEEDINGS OF THE

IECON '94

20th International Conference on Industrial Electronics Control and Instrumentation

September 5-9, 1994
Engineering Faculty Building
University of Bologna
Bologna, Italy

Volume 2 of 3