The Use of the Philips TM1300 for Machine OCR

Imaging systems employed in demanding industrial and military applications, such as computer vision and automatic target recognition, typically require real-time high-performance computing resources. While these systems have traditionally relied on proprietary architectures and custom components, recent advances in high-performance general-purpose microprocessor technology have produced an abundance of low cost components suitable for use in high-performance computing systems.

A common pitfall in the design of high-performance imaging systems, particularly systems employing scalable multiprocessor architectures, is the failure to balance computational and memory bandwidth with I/O. The recent introduction of microprocessors with large internal caches and high-performance external memory interfaces make it practical to design high-performance imaging systems with balanced computational and memory bandwidth. In these systems, both the board level memory and I/O architecture as well as the microprocessor memory have significant performance impact. Systems that do not scale the memory bus bandwidth as processors are added do not typically improve in performance, or they reach a limiting value after some initial small gains in performance. In addition, the I/O bandwidth plays a significant role in overall performance.

Therefore, the most important factors in the selection of a programmable imaging system are: (a) will the system be "native" or will it use a co-processor model, (b) which processor has the correct balance of processing and I/O for the application, and finally (c) what is the overall cost? A real-world example of a design will be presented, and its performance and cost will be analyzed. We conclude by outlining the advantages of the Philips TM1X00 series processors for machine vision applications.

Description of the Application

In order to characterize the advantages of the various state-of-the-art microprocessors, we present the following typical application. We will calculate the performance of the latest microprocessors from Analog Devices, Intel®, Philips Semiconductors and Texas Instruments[1].

The important properties of the selected processors are presented in the table below. Cluster mode designs will reach bus saturation very quickly, to the point that adding additional processors will not improve the rate at which calculations can be performed. The table assumes that the processors are isolated from each other when they are performing the calculations. This assumption is made because most multiprocessor co-processor boards have isolated or local memory designs due to scalability issues. The only non-local memory design considered is the "native" PIII-450 design, which is added for comparison purposes.

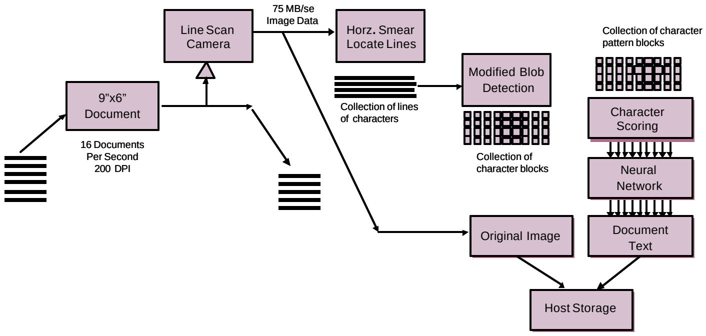
SPECIFICATION	Intel PIII-450	Philips TM1300	TI C6701	ADI 21160
Architecture	CISC	VLIW	VLIW	VLIW
FPU	Yes	Yes	Yes	yes
MFLOPs (Peak)	250	720	1000	600
MIPS (Peak)	450	900	1336	100
MOPS (Peak)	900	4860	1336	800
Memory Bus Bandwidth (MB/s)	800	572	332	400
1K FP cfft (µsec)	300	106	108	90
1K 16 bit cfft (µsec)	191	63	108	90
1K FP dot product (µsec)	7.38	2.84	3.07	5.12
16x16 MACs (MMAC/s)	2.32	1.42	3.07	5.32
8x8 MACs (MMAC/s)	9.34	6.55	7.11	11.80
512^2x8 bit Conv3x3 (msec)	5.34	2.62	7.11	11.80
512^3x8 bit Erosion/Dilation (msec)	7.34	1.42	3.62	3.93
"Glue" Logic Cost ($/CPU)	$150	$3	$65	$39
CPU Price ($)	$300	$100	$150	$100

An OCR Application

The application presented is a typical 'simple' Optical Character Recognition application. The application is considered simple because the template set contains only one font face. The document feed rate is typical of a medium speed application (1000 documents/minute). High speed applications can exceed 2000 documents per minute in extreme cases.

[1] The Motorola PPC750-AV (AltiVec) and the Hitachi/Equator MAP1000 are not presented since, at the time of this writing, there is not sufficient public information available to adequately present them.

[2] & 3 Herman, GT. "Image reconstruction for projections", Academic Press. 1980.



The documents are imaged at 16 per second, or 960 documents per minute, with a line scan camera at 200 pixels per inch. The documents are 9 inches long (parallel to the motion) by 6 inches (perpendicular to the motion or parallel to the camera line).

The image of the document is 1200x1800x8 gray scale pixels. Of these pixels 25 pixels (1/8th of an inch) is a border containing no ink. The document contains 900 characters maximum. The system will accept 7 to 32 point characters. Less than 6 point characters generate false reads, as the document is not imaged at a high enough resolution. It is an application requirement that the computer keep up with the document feeder.

The Data Source

The image data is taken form a 2048 pixel line scan camera. Only one side of the document is imaged. The document moves at 160 inches per second, which results in a line rate of 32 K-lines per second. Camera data is supplied to the computer system as two streams of 8 bit data (two 'taps') at 37.5 MHz.

A two tap camera is used to reduce the clock rate, and ease the signaling between the camera and the computer (LVDS). In this application a single tap camera would generate data at 75 MHz, which is much more difficult to interconnect. Cable routing, impedance, and electromagnetic compatibility are also far more difficult at 75 MHz.

Processing

Locate the Characters (i.e. Segmentation)

As the lines come in, a horizontal smear test finds interline gaps. This is a process in which the pixels for a line are swept from edge to edge, combining them as they go. At the end of the line the interline gap is easily detected as those lines that had no ink on them.

When the top and bottom of a line have been detected, the characters are separated into separate blobs with a blob detection routine that sweeps from left to right over the line. Blobs above and below each other are grouped together.

The characters are placed in bounding boxes which are warped to the template box which is 16 x 16. The bounding boxes for the characters will not be 16 x 16 typically, (e.g. a 20 point character's boarding box would be about 56 pixels tall, and 42 pixels wide).

At this point we have a collection of characters in 16 x 16 pixel boxes that need to be recognized.

Pattern application

The character recognition is accomplished by comparing a group of templates with the characters. The template is compared with the unknown by using the absolute value of the difference between the template pixel and the character pixel. These positive differences are totaled, which is taken as the 'score' for the unknown on the template.

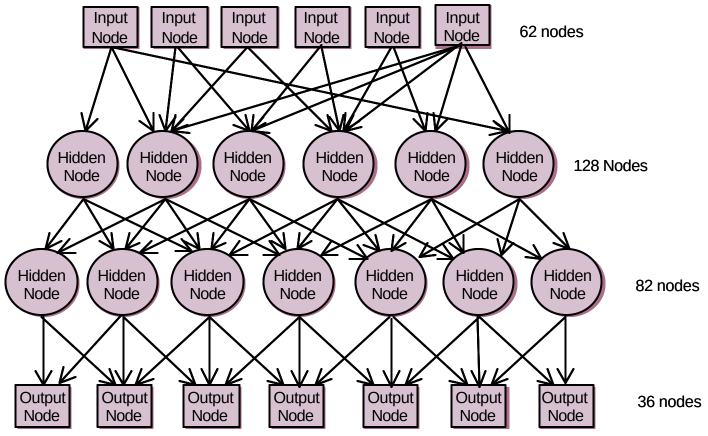
There are 62 templates – 26 for upper case letters, 26 for lower case letters, and 10 for the numbers. Diacritical marks are ignored.

Character selection

The scores from all the templates for an unknown form a feature vector which is applied to a neural network of 4 layers. The node with the highest output, in the output layer of the neural network selects the character that is assigned to the unknown.

The neural network has been pertained and it has 62 nodes in the input layer – each node maps to a particular part of the feature vector (i.e. a template score). The outputs of the input layer are passed to the first hidden layer through the first set of weights, which were determined when the neural network was trained. The first hidden layer has 128 nodes. The first hidden layer passes its outputs to the second hidden layer of 82 nodes in a similar fashion. The output layer processes the output of the second hidden layer and has 36 nodes – 26 for the letters, and 10 for the numbers. A diagram of the neural network is shown in the figure below.

Character Pattern Scores



Character Selection Scores

Data Sinks

Result data is the original image – the character assignments. This data is passed to the host for storage.

Computational Complexity

Horizontal smear takes one pixel read and one compare for the length of the line which is 1150 after the 50 pixels of margin are removed.
The maximum height of a line is 60 pixels. The line of 1150 x 60 is read once, looking for a break in the ink indicating an inter-character gap. This requires 69,000 pixel reads and 69,000 pixel compares.

The characters from gap to gap are transferred to another area, which contains only background (zeros). The bounding box of this character is warped to fit the template box of 16 x 16. This requires 256 weighted sums of 4 elements, or 1024 pixel reads, 256 pixel writes, 1024 multiplies and 768 adds. The warped character is tested against the 62 templates.

For each template, 256 bytes of the template and 256 bytes of the unknown is read in, subtracted, absolute valued, and summed. The resulting number is output to memory. For the 62 templates this works out to 31744 byte reads, 256 byte subtracts, up to 256 negates, 255 adds, one 32 bit write (floating point total).

Next the 62 floating point scores from the template test phase are passed through a previously trained neural network. This requires, 62 x 128 + 128 x 82 + 82 x 36 = 21384 multiplies, the same number of adds, and 128 + 82 + 36 = 246 nonlinear transformations, each requiring a 2 compares, 2 multiplies, and one add, (a piece wise linear function of 16 points).

Computational Complexity for the Whole Document

Horizontal smear takes 216,000 reads, and 216,000 compares (10% of the lines are 'in the gap') for the whole document.

Segmentation of each line takes 69,000 pixel reads and 69,000 compares, producing 30 characters maximum. There are 30 lines on a document, so this works out to 2,070,000 pixel reads, and 2,070,000 pixel compares for the whole document. (Note: the number of lines depends on the height of the characters, but does not change the work load, as more lines of small characters actually have less pixels because of the increase in gaps.)

Taking the worst case of 900 characters:

> Warping takes 900 x (1024 pixel reads, 256 pixel writes, 1024 multiplies, 768 adds) or, 921,600 pixel reads, 230,400 pixel writes, 921,600 multiplies, and 691,200 adds.
>
> For each of the 900 characters in the document, 62 masks need to be tested, or 55,800 mask evaluations. 55,800 x (512 reads, 256 subtracts, 256 compares, 255 adds, one floating point write of 4 bytes), or 28,569,600 pixels reads, 14,284,800 subtracts, 14,284,800 compares, 14,229,000 adds, and 232,200 writes (bytes).
>
> The neural network requires, 900 x (248 reads, 21384 multiplies, 21384 adds, and 246 x (2 compares, 2 multiplies), or 223,200 reads, 19,688,400 multiplies, 19,245,600 adds, and 475,200 compares (this includes the 36 compares per character to select the final character).

Complexity Totals (million-operations)					
Step	**Reads**	**Writes**	**Add/Sub**	**Multi**	**Compares**
Smear	0.216				0.216
Segment	2.070				2.070
Warp	0.922	0.231	0.691	0.921	
Mask	28.570	0.232	28.513		14.284
Net	0.233	19.689	19.245	19.688	0.475
Total=135	32.011	20.152	48.449	19.688	14.759

Theoretical Maximum Performance

The total number of operations performed by the processor is the sum of the total row – we include the read and writes because a slot is used to issue those functions. The total is 135.06 million operations. The TM1300 processor operates at 180 MHz instruction rate, and most of the operations are integer so they can be issued from any of the 5 issue slots. Assuming perfect packing we get 0.150 seconds to process the page (at best) (135.06 / 180 / 5 = 0.150 sec)

The TI TMS320C6701 operates at 166 MHz and is able to start up to 4 floating point operations per clock. Again, assuming perfect packing of instructions, the C6701 will process one document in 0.124 sec. On the C6701 the only way to obtain high performance is to use the DMA controller to bring data into the internal SRAM, and only process data from SRAM. If the processor reads external memory directly it slows down by a factor of 8. The DMA activity will contend with the processor for access to the internal SRAM, which will extend the processing time per page to 0.254 seconds per page.

The Analog Devices ADSP21160 will operate at about 100 MHz internal and 50 MHz externally. The processor is able to perform up to 4 floating point operations per clock. Instruction packing is not a significant problem on the 21160, nor is access to internal SRAM because of the dual porting. Again, assuming the difficult buffer management problem is solved, the 21160 should be able to perform the computational part of the problem in 0.414 seconds. The time to transfer the data from external memory is 0.130 seconds, so the application is processor limited.

The Intel PIII-450 operates at 450 MHz internally with a 100 MHz bus speed. Data transfer time for external memory in a single processor system is 0.0652 seconds. The PIII is not a VLIW processor and can only produce one result on every clock. A single processor can process the page in 0.249 sec. Two processors can process two pages in 0.314 sec, the time goes up slightly due to bus contention. Three processors process three pages in 0.380 sec. Finally, four processors can process four pages in 0.445 sec.

Selecting the Number of Processors

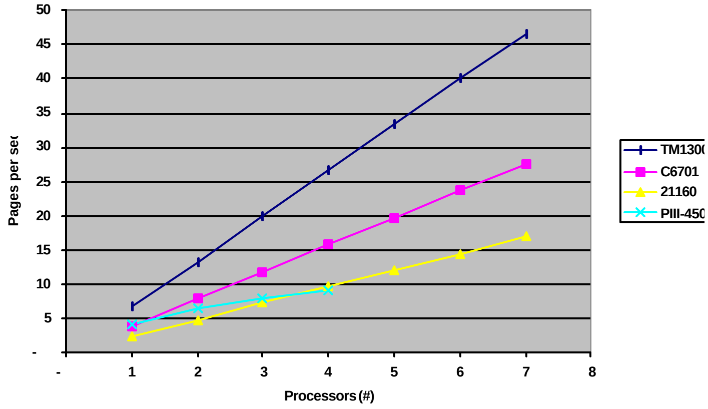
- To keep up with 16 pages per second we need a through-put of 0.0625 sec/page.
- Three TM1300 processors will process 3 pages in 0.150 seconds, or 0.050 sec/page.
- Four TMS320C6701 process 4 pages in 0.0635 sec/page – not quite fast enough but considering the accuracy of these estimates, four will probably work.
- Seven AD21060 process 7 pages in 0.0591 sec/page.
- Finally the P5 processor can not reach the goal of 0.062 sec/page – one processor requires 0.249 sec/page, two 0.157 sec/page, three 0.126 sec/page, and four 0.111 sec/page. The contention for memory prevents the PIII from reaching the goal. If the PIII were available as isolated processors and memory, it would require four processors to perform the task.

Data Flow Analysis

- 2.16 megabytes per page is read in from the camera.
- 52.163 megabytes is read and written per page, during processing
- 2.16 megabytes are output to the host with each page
- Total memory traffic is 56.483 megabytes per page
- The TM1300(s) memory bus will be 10% loaded
- The C6701(s) external memory bus will be 14% loaded
- The AD21160(s) external memory bus will be 14% loaded
- The PIII(s) memory bus is 105% loaded at 4 processors

Scaling the Application to Meet the Requirements

- In this application we found that processing and memory architecture is the most important factor in determining the size of the system.
- The PCI bus is 56% loaded by image traffic to the host.



Conclusions

An example application has been presented illustrating how memory bandwidth and processing through-put limits the performance of multiprocessor system. Whereas memory bus saturation severely limits the scalability of cluster based architectures (i.e. PIII-450), local memory architectures allow throughput to scale linearly with the number of processors. In addition, VLIW processors like the TM1300 and the TMS320C6701 can make up for their lower clock frequencies by supplying multiple processing elements.

Notably, an excellent processor, the Intel PIII, is limited by its surrounding logic (the PC) and is unable to perform in this application. Therefore the most practical solution for demanding application remains a co-processor board that is more scalable, has higher throughput, and ultimately is cheaper than the native solution. Among these, the Philips TM1300 stands out in the performance versus price curves.

*Correspondence: Email; joe@alacron.com; http://www.alacron.com; Fax (603) 891-2750

ALACRON

71 Spit Brook Road, Suite 200
Nashua, NH 03060
Tel: 603.891.2750 Fax: 603.891.2745
Web: www.alacron.com E-mail: sales@alacron.com