The Use of the Philips TM1X00 for Machine Vision

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? Real-world examples of such designs will be presented, and their performance and cost will be analyzed. We conclude by outlining the advantages of the Philips TM1X00 series processors for machine vision applications.

Description of Applications

In order to characterize the advantages of the various state-of-the-art microprocessors, we will present three typical applications. Included in these applications is a high I/O application such as wafer mask inspection, a balanced I/O and computation application such as fruit inspection, and finally a demanding I/O and computation application such as ultrasound. We will calculate the performance of the latest microprocessors from Analog Devices, Intel®, Philips Semiconductors and Texas Instruments[1].

[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.

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 is the "native" PIII-450 design, which is added for comparison purposes.

SPECIFICATION	IntelPIII-450	Philips TM1300	TI C6701	ADI 21160
Architecture	CISC	VLIW	VLIW	VLIW
FPU	Yes	Yes	Yes	yes
MFLOPs (Peak)	250	720	1000	600
16x16 MACs (MMAC/s)	450	360	334	200
8x8 MACs (MMAC/s)	450	1440	334	200
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
1K 16 bit dot product (µsec)	2.32	1.42	3.07	5.32
512^2xFP Conv3x3 (msec)	9.34	6.55	7.11	11.80
512^2x8 bit Conv3x3 (msec)	5.34	2.62	7.11	11.80
512^2x8 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

Mask Inspection

The following outlines the inspection of masks used in semiconductor (integrated circuit) manufacturing. This is a difficult application due to the resolution required to detect defects that are significant to the process. The small feature size (less than 1 micron) used in today's processing exacerbates the problem because of the squared relationship between resolution

and the number of pixels to be processed, which is in direct relationship with the time needed to complete the inspection.

The equipment used encompasses three precision elements; (a) a digital line-scan camera, which in this application was 4096 pixels long, (b) a high quality lens able to image over the length of the line scan sensor at the diffraction limit of the light being used, and finally (c) a positioning stage which moves the mask being inspected under the sensor in successive passes. This last step is performed with enough precision to be sure that every part of the mask is imaged at the required resolution.
Lighting and lens selection are not trivial as 0.5 microns approaches ultra-violet wavelengths.

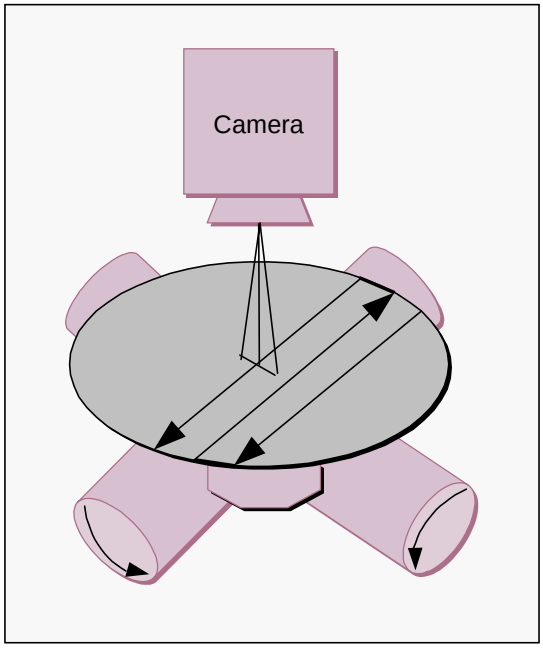


Diagram of Inspection Apparatus

Imaging the Mask

The mask is imaged by moving it under the line scan camera in a back and forth pattern, so that each pass overlaps the last pass by 1% (40 pixels, 20 microns). This assures that no portion of the mask is left unexamined due to positioning errors of the stage. It also gives the processing system enough data to determine if a defect is bridging two passes, or if two defects are near the edge of the two passes.

The lighting of the mask is designed so that the background is dark and any defects will show as bright spots (i.e. Dark Field Lighting). This is typically accomplished by lighting from the sides so that any defect will scatter light into the camera.

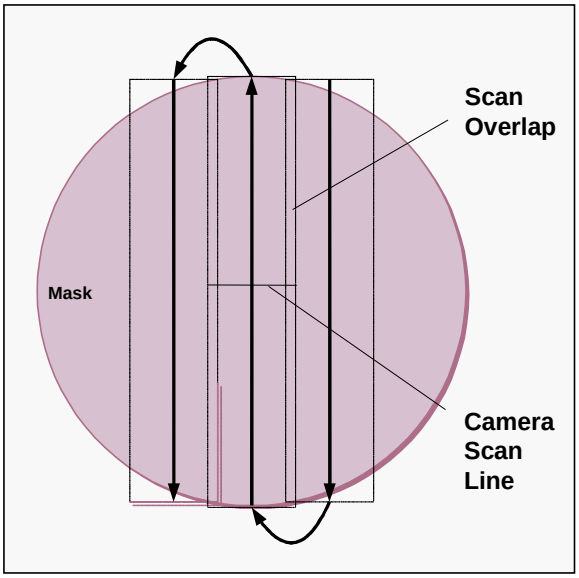


Diagram of Imaging Path

As each pass is executed, image data is processed by the computer system, which retains images of the defects, throwing out normal background data. This discarding of background data is a cost saving method, as it eliminates the need to store 40 GB of image data, most of which would be uninteresting. In addition, the data is being collected at a high data rate (from 50 to 200 Mbytes/s), which would require expensive hardware to capture and store (striping RAID disk systems are used typically).

Image Processing

Image processing consists of two parts—one very simple step which must be performed on every pixel, and more complex image processing to be performed on the defect images only.

The first part of the processing corrects the pixels for variation in the line scan detector (gain and dark current correction), and then compares the corrected pixel to a threshold. The correction step allows the threshold to be set very low so that a greater portion of the defect is detected for later processing. Also, correcting the sensor for a 'flat field' of view will reduce systematic errors, which might show up as errors that are correlated to the position of the stage and camera, rather than true defects in the mask.

The second part of the processing collects the image data for the defect by collecting pixels in a rectangular region that is slightly larger than the defect image. These regions of interest (ROIs) are collected and further processed by a blob detection algorithm, and then measured. The measurements taken are position, area of the convex hull of the defect, radius of the smallest circle that will encompass the defect, average density of the defect (brightness), and the perimeter of the defect. This data will be processed by the host computer to determine if the mask should be discarded. The defects are utilized by later

processes that use the mask to reject circuits that are produced by portions of a defective mask, or to trigger a cleaning step should the type of defect indicated suggest contamination.

Performance

The processors shown in the table below were compared in the implementation of this application. In each case the central loop of the first processing step dictated the number of processors needed to 'keep up' with processing the pixels as they came in. Additional processing power is used in defect analysis, but this step is performed after the imaging is complete, removing the high-performance requirement.

The application is parallelized by data partitioning so that each processor gets a portion of the data, a vertical slice along the motion of the stage. Each vertical slice is taken to overlap (40 pixels) so that bridging defects can be resolved. Errors are collected in memory and processed at the end of the scan, as explained above.

There is no physical process that dictates the speed at which the image should be collected.[2] It is a trade off between the cost of the system required to inspect the mask and the cost of the time it takes in the process.[3] The first step in determining the cost is determining the number of processors required to inspect the mask in a given amount of time. For the purposes of this article, 60 minutes down to 15 minutes is considered. At higher performances (>200 Mbyte/s) two cameras are required as the data rate exceeds that obtainable from the faster line scan cameras (Dalsa CT-F3-4096 8 tap camera).

As can be seen in the table below, the number of processors required is best for the TriMedia TM1300 processor. The TI processors came in second, however, they will not fare too well when cost is considered. The TriMedia TM1100 processor also makes a good showing. The ADI processors do not stack up as well due to the limited number of processing units available in each processor. The SHARC 21060 is shown as a reference point.

Data Rate	200 MB/s	160 MB/s	100 MB/s	80 MB/s	50 MB/s
Processor	15 min	20 Min	30 Min	40 Min	60 Min
ADI ADSP21160	8	6	4	3	2
Philips TM1300	3	2	1	1	1
TI TMS320C620X	4	3	2	2	1
Intel-PIII-450	NP	NP	1	1	1

[2] Actually this is not true; if the scan takes too long, diurnal thermal variations in the apparatus become significant, and then there's always rust…

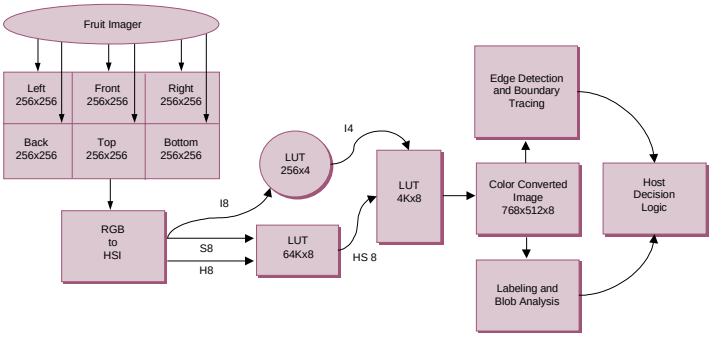
3 Also the patience of the operator.

The PIII-450 processor almost outperforms the other processors, but it is limited by the mother board's I/O capability (PCI) of less than 132 MB/s. If one were to build a private memory multiprocessor PIII based product, it would perform quite well in the kind of application requiring high memory bandwidths, since its memory system is currently the fastest (800 MB/s peak). When cost is considered, the PIII processor does not fare as well—the low cost versions (<330 MHz) do not perform as well as other processors in the table, while the higher performance parts (>400MHz) perform well but are expensive. In addition, the physical size of the high-performance Pentium processors make them difficult to use.

The mask inspection application is an example of a memory performance limited application, which is solved by scaling the memory bandwidth to deliver the highest performance solution with tolerable cost. Real-time constrained applications exhibit other characteristics as the next example shows.

A Real-Time Inspection Application for Fruit

The figure below outlines a real-time inspection system for fruit. To measure the quality of fruit, the colors and shapes are used. The fruit is imaged with a color digital imaging system, the images are analyzed, and results reported to a host computer to determine the final disposition of each piece of fruit. The fruit is presented to the imaging system at a rate of 10 per second. It is a design requirement that all the fruit be inspected.



Fruit Inspection

Collecting the Image

During this phase of the application the fruit is imaged by a camera that captures the fruit from six directions – front, left, back, right, top, and bottom – and supplies these images to the computer system as six separate images, each at 256x256 pixels. The six images are merged into one image of 768x512 for ease of handling, three images across and two down. This image is digitized as 24-bit RGB data.

Color Evaluation

The image is color space converted from RGB to HSI (hue, saturation and intensity). The three 8-bit values are converted to

one 8-bit value by converting the 'I' value to a 4-bit value via a look up table (LUT), then converting the H and S values to an 8-bit value via another LUT. The 12-bit value formed from the 4-bit intensity value and the 8-bit color value is converted to an 8-bit value via a final LUT. (In principle this could be done with one LUT 24 million bytes deep.)

The LUTs are precomputed to allow the color differences in the fruit that are significant to its grading to be easily detected. After the RGB pixel values are converted to the 8-bit pixels, a blob analysis is run on the image. Regions of constant color (8-bit value) are labeled in each of the 6 views. A table of regions is reported to the host computer containing the location of the blob, its size, color and bounding box. These values are used by the host computer to determine if the fruit is ripe, over-ripe, or damaged.

Shape Evaluation

Shape evaluation is performed in each view by determining the perimeter of the fruit in each view, the area enclosed by the perimeter, the convex hull of the perimeter, and the area enclosed by the convex hull. The host computer uses the convex hull and the perimeter to determine if a piece of the fruit is missing or if the fruit is oddly shaped. For example, in the case of spherical fruit, the convex hull and the perimeter should be nearly the same. However, for a banana, the convex hull will enclose more area than the perimeter.

Computational and Bus Bandwidth Requirements

The table below shows the counts of the number of operations and usage of memory for the selected processors. The number of processors required can be determined by computing the time it would take one processor to do the processing, and dividing that by the time actually available.

Operations	CPU Ops	BUS Bytes-R	BUS Bytes-W	I/O Bytes
Acquire front			786,432	786,432
Acquire left			786,432	786,432
Acquire back			786,432	786,432
Acquire right			786,432	786,432
Acquire top			786,432	786,432
Acquire bottom			786,432	786,432
Calibration		4,718,592		
Stitch image together	4,718,592		1,179,648	
Convert from RGB to HSI	17,694,720	1,179,648		
Convert HSI to color index	7,077,888		393,216	
Blob analysis	524,288	786,432	786,432	
Inspect Shape	866,304	430,080	720	720
Report enclosed area	786,432	393,216	240	240
Report perimeter length	64,512	21,504	240	240
Report convex hull area	15,360	15,360	240	240
Entire Application	30,881,792	7,114,752	7,078,608	4,719,312

Inspection Results

In this application the same set of processors is compared, but this time their computational performance is significant. The performance of the processors is shown in the table below.

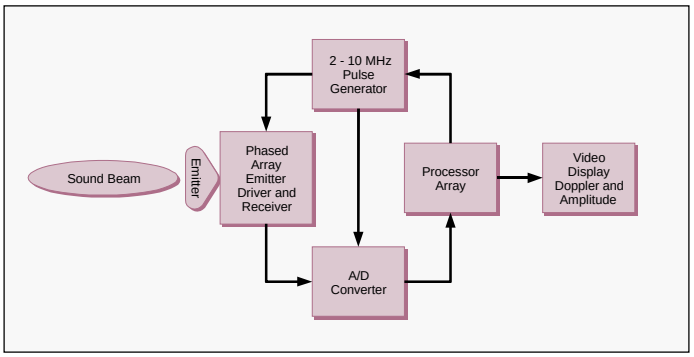
When the need to process 10 fruits per second is considered, the total operations and memory bus cycles dictate the number of processors required. In this application the PIII processor is able to participate as the data rate is below its 132 MB/s limitation.

Intel PIII-450	Philips TM1300	TI C6701	ADI 21160
2	1	3	2

Number of processors required for fruit inspection application.

Ultrasound Application

Processing power available from many DSP processors has allowed a reduction of the amount of analog processing required in Ultrasound applications. In addition, they have reached performance levels that will allow additional features to be implemented without a significant increase in cost.



Block Diagram of Ultrasound System

Visual and auditory feedback are critical to the placement of the probe, which is extremely important to the quality of the test results. To assist in this difficult task, the ultrasound system must process the data it collects and render an image as quickly as possible. This would allow the technician to receive essential

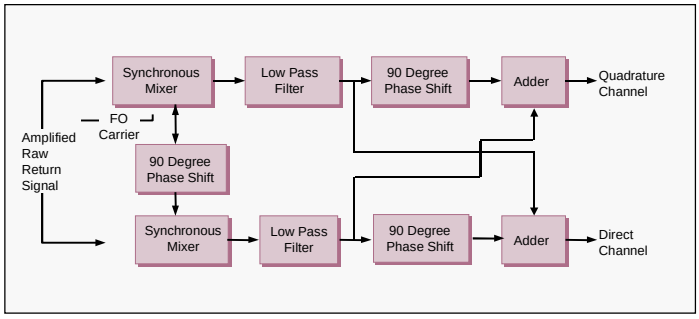
visual feedback in the placement of the ultrasound probe without significant lag, which would make the positioning difficult. This requirement generates a tight latency requirement on the processing of ultrasound data.

A typical ultrasound machine might take data at 20 million 12-bit samples-per-second on two channels. This rate allows operation at carrier frequencies of 10 MHz or less, which is typical for ultrasound systems.

Several processing steps must be performed on this high sample rate data to reliably detect the signals generated by the sound returns. Two components of the signal are of interest – the **amplitude of the reflection** and its **frequency**. The amplitude of the return is used to detect density of tissue, while the frequency is used to detect motion via the Doppler effect.

The emitted sound beam is generated from a phased array emitter, which is pulsed with a carrier frequency of from 2 to 10 megahertz. The phased array emitter allows the sound beam to be positioned electrically ('beam steering') and focused into a small area ('beam forming'). The sound reflected back is received by the emitter and passed to an A-D converter for conversion to digital form. The A-D converter processes the return signal and the original emitted signal generating two streams of digital data—a signal and a reference channel.

The digital data is processed in two ways. First, the amplitude of the signal return is demodulated from the high frequency carrier with a synchronous AM detector. Second, the phase of the returns are compared to the originally emitted carrier, with a synchronous quadrature detector. This phase measurement is combined with other phase measurements to detect the Doppler shift in the return signal. In addition to these demodulation steps, the signals are gated—that is, selected by time and angle—to report data from a region selected by the operator. Additional processing can be performed to remove artifacts such as echo returns, variable sound velocities, and geometrical artifacts such as off angle Doppler measurements.



The quadrature phase detector is implemented in software on the processor array. The pair of synchronous mixers and the low-pass-filters operate at the 20MHz sample rate. The output of the low-pass-filters is sample rate converted down to a 25 KHz for further processing (see diagram above).

To implement the high sample rate processing, the following steps are required: a multiply in each synchronous mixer; a delay for the 90 degree phase shift; and a multistage rate converting low pass filter. This filter is implemented as a three tap filter, followed by a rate conversion down to 1MHz, then another three tap filter and a rate conversion down to 100KHz, then a 17 tap filter and a rate conversion down to 25 KHz. The rate conversions reduce the computational load and allow a sharp cutoff filter implementation. A non-rate converting filter would require significantly greater processing bandwidth. Each filter tap involves a multiply and an add. The phase shifts are essentially free as they are implemented through modifying the address from which the data is taken. Totaling the computational requirement we have 8 multiplies and 6 adds at 20 MHz, then 6 multiplies and 6 adds at 1 MHz, then 34 multiplies and 34 adds at 100 KHz. The required performance is (280+12+6.8=298.8) about 300 Mflops. This is well within the reach of DSP processors on the market today.

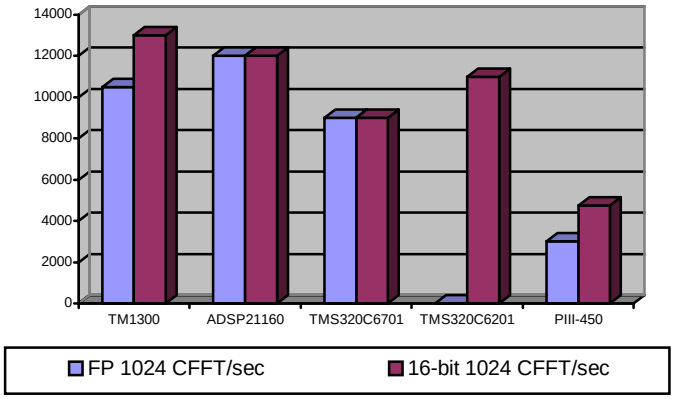
After this processing, the two channels are combined to form the Doppler shift signal, which is gated and then processed with an FFT for display. The gating selects a window determined by the operator in which signals are taken and out of which the signals are ignored. This involves varying the data selection window, which translates into a start and stop time, for each beam position.

The demodulated amplitude signal is corrected by applying an increasing gain with time to correct for attenuation, and warped to correct for speed variations. The amplitude signal is plotted on the display at the locations dictated by the beam position. This plot creates the image of the area being studied.

These post processing steps, except for the FFTs, are insignificant when compared to the processing needed by the demodulator and the FFTs. The FFT processing is performed to display the spectrum of the Doppler return, which translates directly into velocity, after considerations for geometry. The character of the spectrum (noisy, smooth, wide band, narrow band) is an indication of the condition of the area being examined.

The processing required by the FFTs depends on the rates selected by the operator. The 25 KHz Doppler signal can be rate converted up to give finer frequency velocity resolution, and can be performed lapped (i.e., using overlapping sliding buffers) to improve low velocity sensitivity of the system. All of these effects translate in an equivalent number of 1K complex FFTs per second. The table and graph below show how well various processors perform the FFT processing.

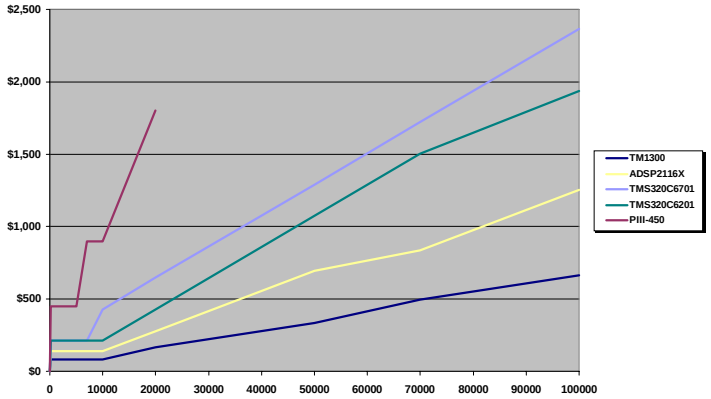
Two considerations are significant to this processing: the rate at which data is passed to the process performing the FFTs, and the bus architecture used. As the number of FFTs per second increase, the computational demand on the processor increases. At the same time, the demands of the processor for data increase as well.



This results in the following requirements for processors assuming almost linear scaling. This is true for most designs except the "native" PIII-450 case.

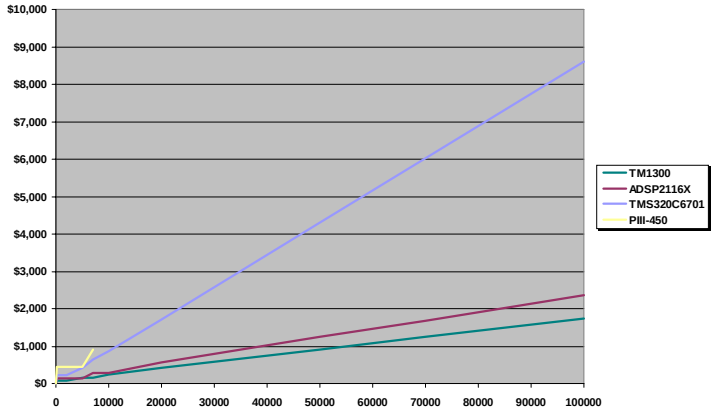
NUMBER OF PROCESSORS REQUIRED					
16 bit 1024 CFFT/sec	**TM1300**	**ADSP 21160**	**TMS 320C6701**	**TMS 320C6201**	**PIII-450**
100	1	1	1	1	1
200	1	1	1	1	1
500	1	1	1	1	1
700	1	1	1	1	1
1000	1	1	1	1	1
2000	1	1	1	1	1
5000	1	1	1	1	1
7000	1	1	1	1	2
10000	1	1	2	1	2
20000	2	2	3	2	4
50000	4	5	6	5	NP
70000	6	6	8	7	NP
100000	8	9	11	9	NP

The cost curves are illustrated in the following graph.



NUMBER OF PROCESSORS REQUIRED				
FP 1024 CFFT/sec	**TM1300**	**ADSP 2116X**	**TMS 320C6701**	**PIII-450**
0	0	0	0	0
100	1	1	1	1
200	1	1	1	1
500	1	1	1	1
700	1	1	1	1
1000	1	1	1	1
2000	1	1	1	1
5000	2	1	2	2
7000	2	2	3	3
10000	3	2	4	NP
20000	5	4	8	NP
50000	11	9	20	NP
70000	15	12	28	NP
100000	21	17	40	NP

The cost curves are illustrated in the following graph.



Thus the TM1300 is the most efficient when cost of the processor and associated glue logic is factored with the number of units needed.

Conclusions

Example applications have been presented illustrating how memory bandwidth limits the performance of multiprocessor systems. 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.

Notably, an excellent processor, the Intel PIII, is limited by its surrounding logic (the PC) and is unable to perform in some applications. Although the use of the AGP bus would improve the situation, its SMP design will ultimately limit its scalability. The most practical solution for demanding application remains a co-processor board, which 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.

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