4PM 5.0

1. Introduction

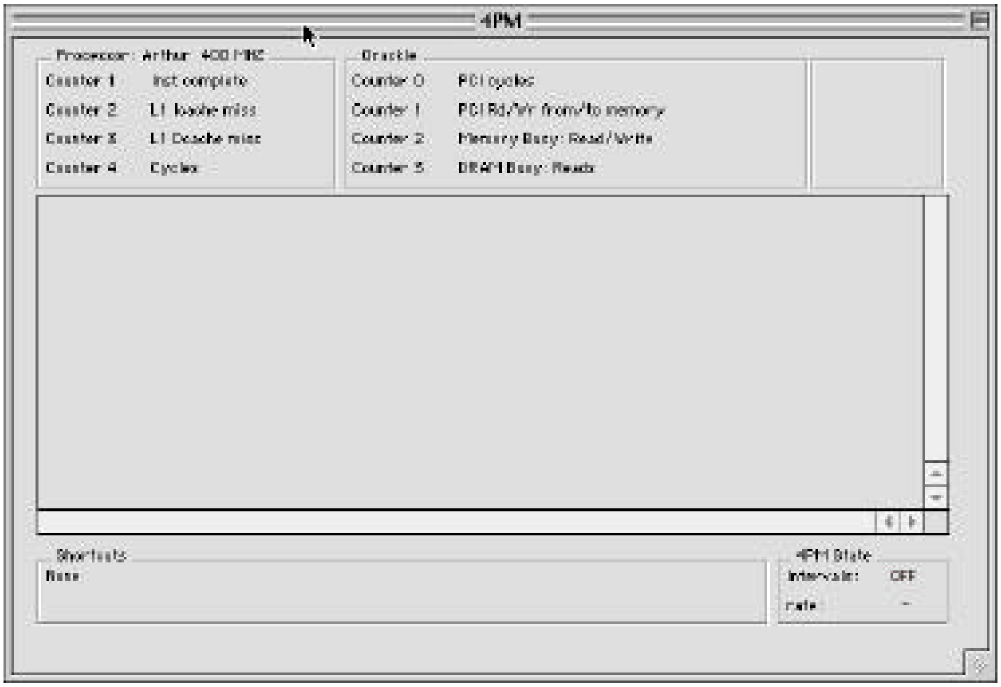
4PM is a program that can be used to control and access the performance monitoring features of the PowerPC 604, G3 and G4 microprocessors as well as the performance monitoring features of the MPC106 memory controller. These counters count events such as instructions executed, TLB and cache misses and many others.

Data from the counters can be combined to report higher level metrics such as memory bandwidth, L1 or L2 cache miss ratios, cpi and others. 4PM can be configured to report counts of events as well as to use multiple counters and report some of these higher level metrics.

4PM provides two mechanisms for starting and stopping the counters: a library which contains start and stop calls and a 'hot key' which toggles the counters on and off. 4PM can report the data as a total count or as a histogram. The data can be saved and read into a spreadsheet.

2. The Interface

The main 4PM window is shown in figure 1. The processor type and speed is shown in the upper left box along with the events selected for the processor counters. The memory controller type is shown in the upper middle box along with the events selected for the four memory controller counters.



Processor: Arthur 400 MHz

Counter 1 Inst complete
Counter 2 L1 Icache miss
Counter 3 L1 Dcache miss
Counter 4 Cycles

Grackle

Counter 0 PCI cycles
Counter 1 PCI Rd/Wr from/to memory
Counter 2 Memory Busy: Read/Write
Counter 3 DRAM Busy: Reads

Shortcuts

None

4PM State

Intervals: OFF
rate –

Figure 1. Main 4PM Window

Any selected shortcuts are shown in the box on the bottom left and the '4PM State' box shows if you are sampling at a time interval.

Configuration options are found by selecting *Preferences* from the *Control* menu shown in figure 2. This brings up the preferences dialog.

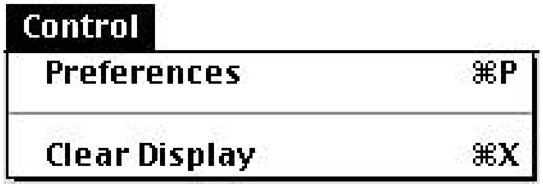
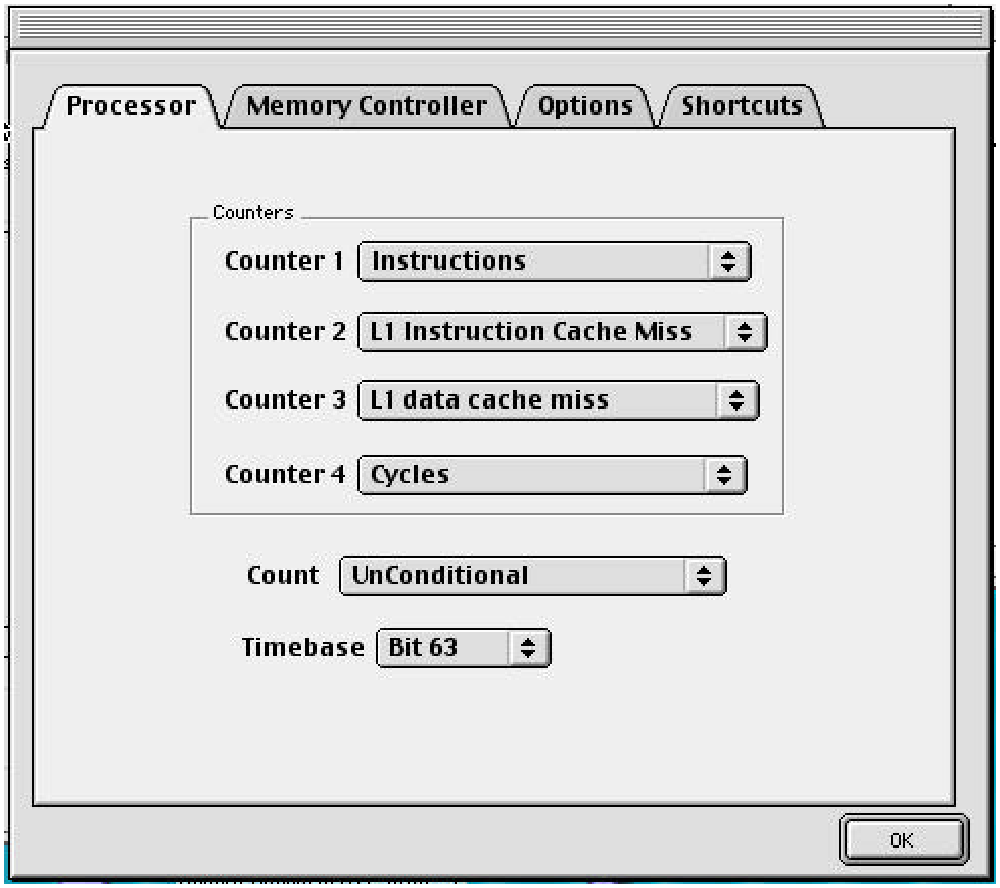


Figure 2. Control Menu

The preferences dialog is shown in figure 3. It has four tabs labeled *Processor*, *Memory Controller*, *Options* and *Shortcuts*. The *Processor* tab is selected in figure 3. When the *Processor* tab is selected, processor events can be chosen for the processor counters by using the popup menu for each counter.



Counters

Counter 1 Instructions

Counter 2 L1 Instruction Cache Miss

Counter 3 L1 data cache miss

Counter 4 Cycles

Count UnConditional

Timebase Bit 63

Figure 3. Processor Tab on Preferences Dialog

The *Count* popup menu on the *Processor* tab selects the state in which to count events. This menu is shown in figure 4. Selecting *Unconditional* (the default) counts all events. Selecting *Supervisor Only* or *User Only* will count only events occurring while the processor is in supervisor or user mode respectively. The states called 'Marked' in the menu refer to the fact that the OS can 'mark' areas of memory. In the current OS the Emulator is marked. Slightly more detailed documentation on processor events and counting state can be found in chapter 11 of the PowerPC 750 User's Manual downloadable at:

http://www.mot.com/SPS/PowerPC/products/semiconductor/cpu/750.html

Counter events are described starting on page 11-7 and states are described on page 11-11. Of course, if you are not using a PowerPC 750 you need to select the appropriate chip's user manual although the information on processor state will probably be the same.

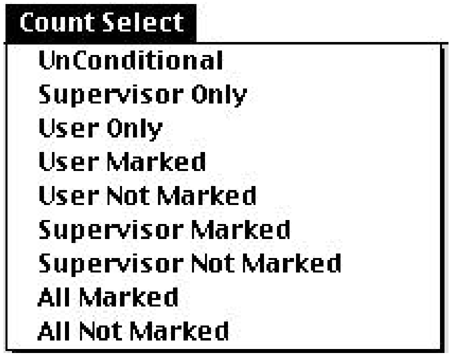
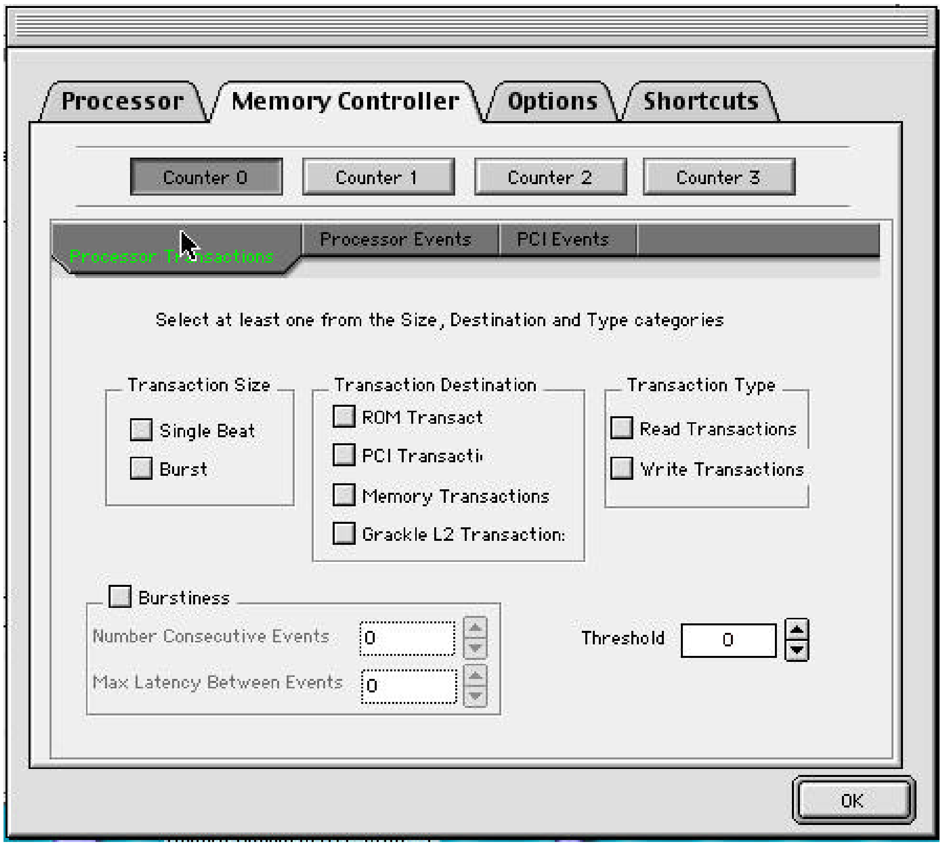


Figure 4. *Count* Popup Menu on the *Processor* Tab

The *Memory Controller* tab is selected to choose memory controller events. Memory controller events are broken up into four categories, *Processor Transactions*, *Processor Events*, *PCI Events* and *Memory Events*. A sub panel in the *Memory Controller* tab represents each of these four categories. The sub panel is selected by clicking on the desired tab in the selection bar (under the cursor in figures 5 through 8). Figure 5 shows the *Processor Transaction* sub panel. Counters 0 and 1 can count events from the first second and third category of memory controller events. Counters 2 and 3 can count events from the second third and fourth categories.



Select at least one from the Size, Destination and Type categories

Transaction Size
□ Single Beat
□ Burst

Transaction Destination
□ ROM Transact
□ PCI Transacti
□ Memory Transactions
□ Grackle L2 Transaction:

Transaction Type
□ Read Transactions
□ Write Transactions

□ Burstiness
Number Consecutive Events 0
Max Latency Between Events 0

Threshold 0

Figure 5. Processor Transaction Selection of Memory Controller Tab

The *Processor Transaction* sub panel allows you to count single beat and/or burst transactions from the processor directed at any or all of four destinations, ROM, PCI, Memory or an external L2. For current Macs there shouldn't be any ROM or GrackleL2 transactions so selecting them may be a waste of time. The type of transaction (read and/or write) should also be selected.

The value in the threshold edit field is the number of bus cycles that transaction time must be greater than to be recorded. For example if 10 were entered in the threshold field and 'Burst', 'Memory Transactions' and 'Read Transactions' were checked the counter will count any burst reads to memory which take more than 10 bus cycles to complete.

To monitor the burstiness of requests, check the 'Burstiness' box and enter values in the two edit fields. Burstiness uses both counter 0 and counter 1 of the memory controller so when it is selected counter 1 can not be configures. When the counters are configured for burstiness counter 0 counts the number of times there are at least X transactions in a row with an acceptable latency of Y bus cycles between any two transactions. For example if you check the 'Burstiness' box and enter 500 in the 'Number Consecutive Events' box and 14 in the 'Max Latency Between Events' box and check the 'Burst', 'Memory Transactions' and 'Read Transactions' check boxes the counter will report the number of times there are 500 burst reads to memory in a row each with less than or equal to 14 cycles of latency between them.

Figure 6 shows the *Processor Events* sub panel. Slightly more detailed descriptions of the events, threshold and burstiness can be found in the addendum to the MPC106 user's manual available for download at Motorola's web site at:

http://www.mot.com/SPS/PowerPC/products/semiconductor/support_chips/106.html

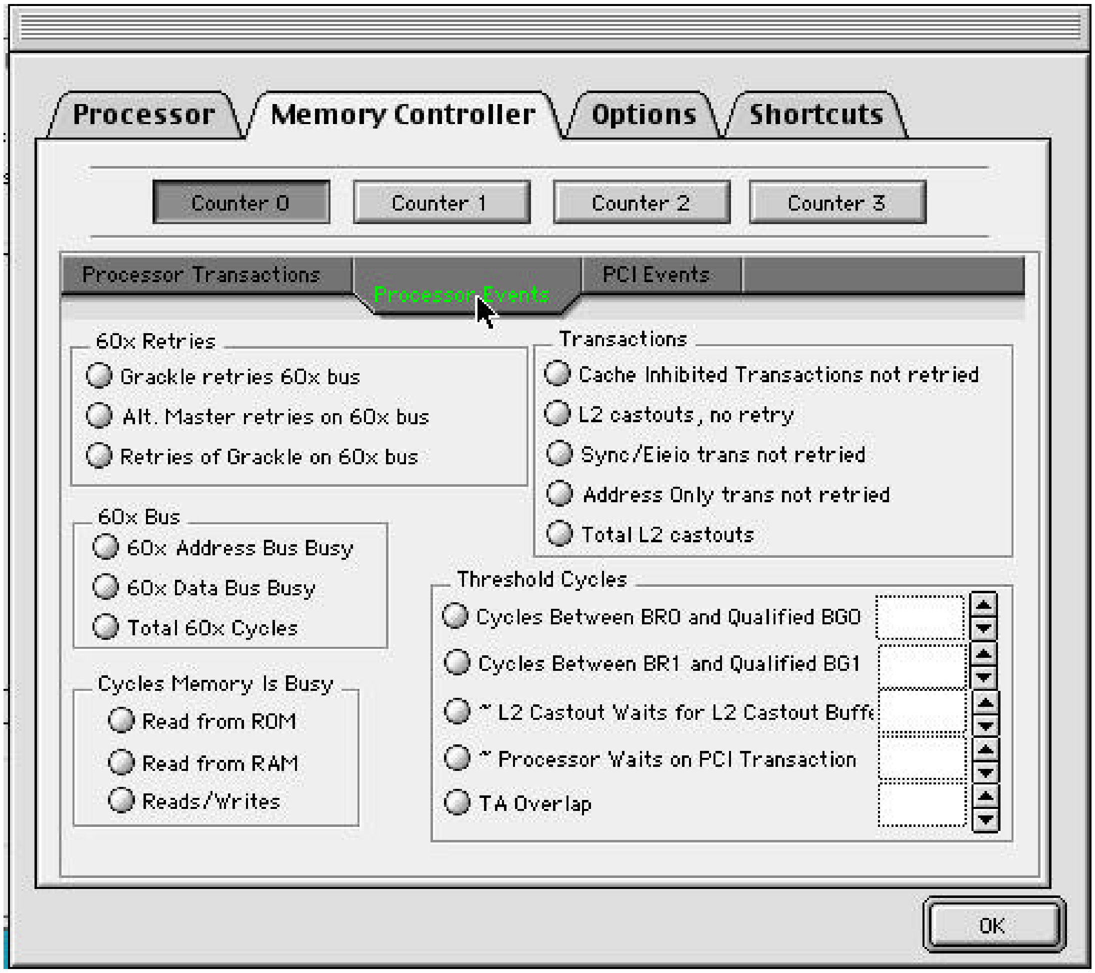
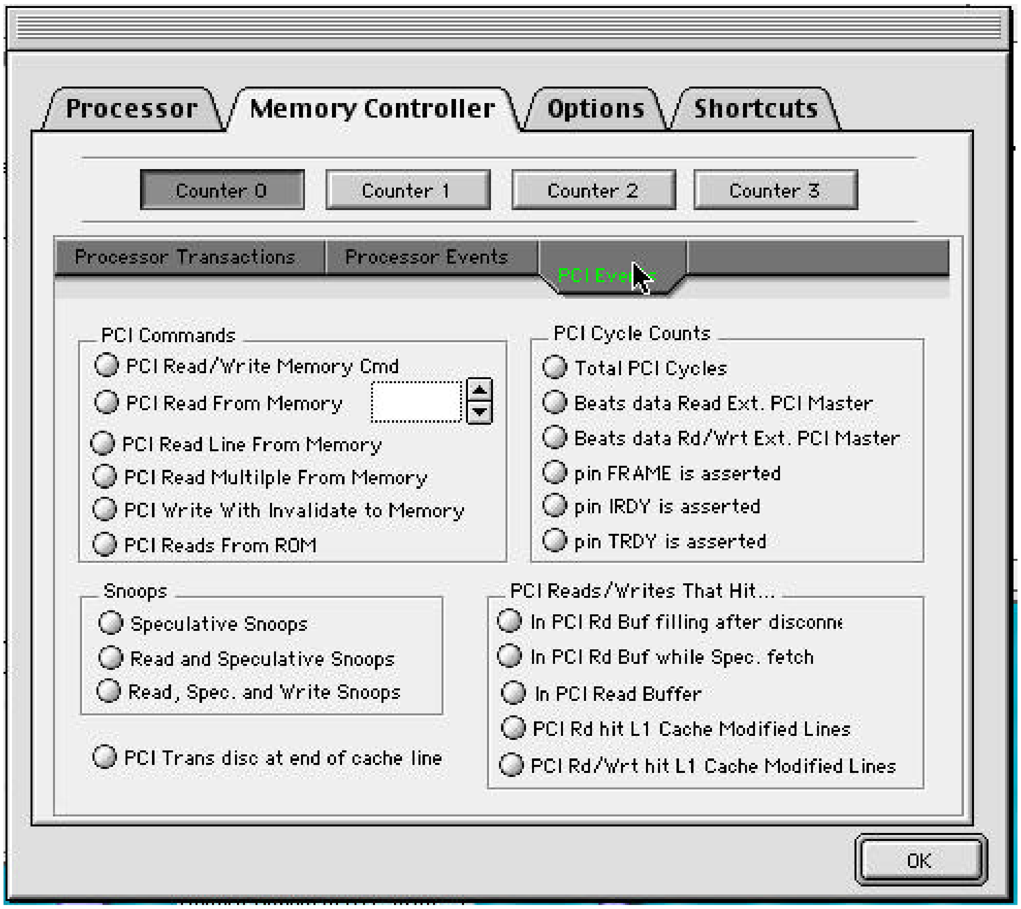


Figure 6. Processor Events Selection of Memory Controller Tab

Figure 7 shows the *PCI Events* sub pane. The 'PCI Read From Memory' event has a text entry field in which a threshold can be entered. When recording this event, the counter increments when the number of cycles from the start of the transaction until the first data is greater than the threshold value. Zero is a valid value and counts all PCI read from memory commands.



PCI Commands

○ PCI Read/Write Memory Cmd
○ PCI Read From Memory
○ PCI Read Line From Memory
○ PCI Read Multilple From Memory
○ PCI Write With Invalidate to Memory
○ PCI Reads From ROM

PCI Cycle Counts

○ Total PCI Cycles
○ Beats data Read Ext. PCI Master
○ Beats data Rd/Wrt Ext. PCI Master
○ pin FRAME is asserted
○ pin IRDY is asserted
○ pin TRDY is asserted

Snoops

○ Speculative Snoops
○ Read and Speculative Snoops
○ Read, Spec. and Write Snoops

○ PCI Trans disc at end of cache line

PCI Reads/Writes That Hit...

○ In PCI Rd Buf filling after disconne
○ In PCI Rd Buf while Spec. fetch
○ In PCI Read Buffer
○ PCI Rd hit L1 Cache Modified Lines
○ PCI Rd/Wrt hit L1 Cache Modified Lines

Figure 7. PCI Events Selection of Memory Controller Tab

Figure 8 shows the *Memory Events* sub pane. Using this pane you can select memory related memory controller events.

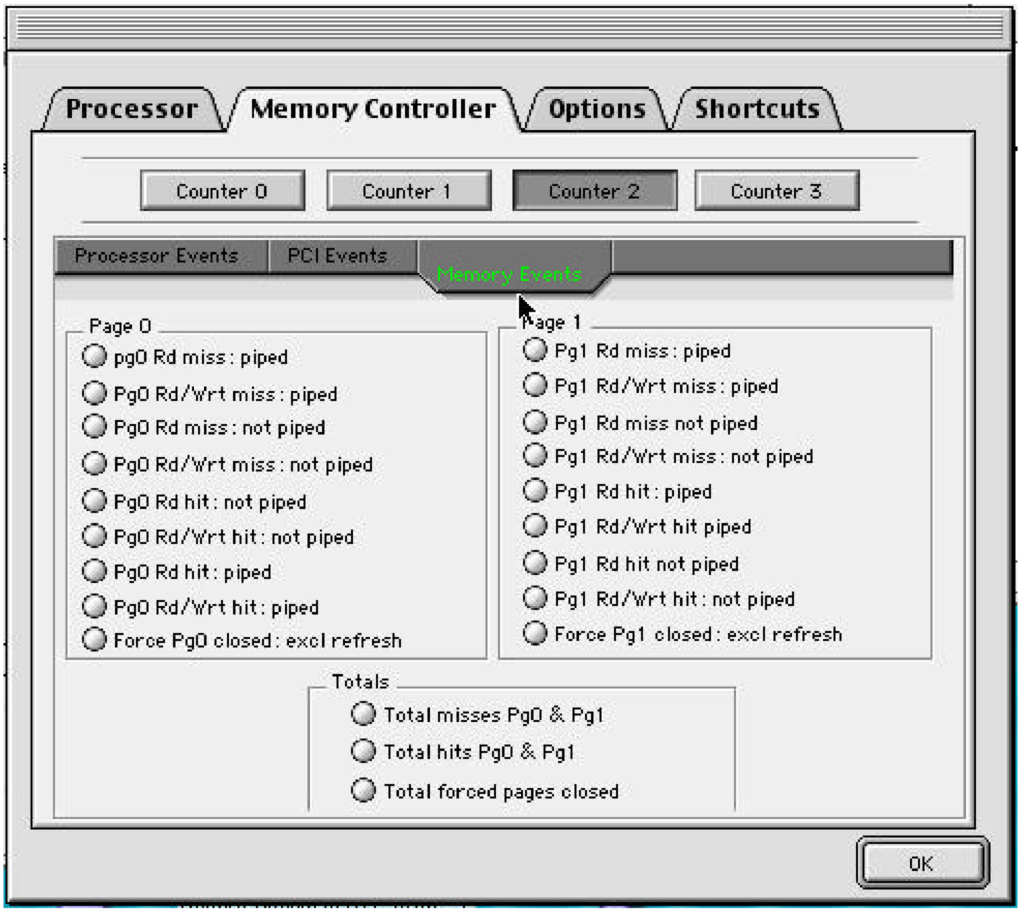
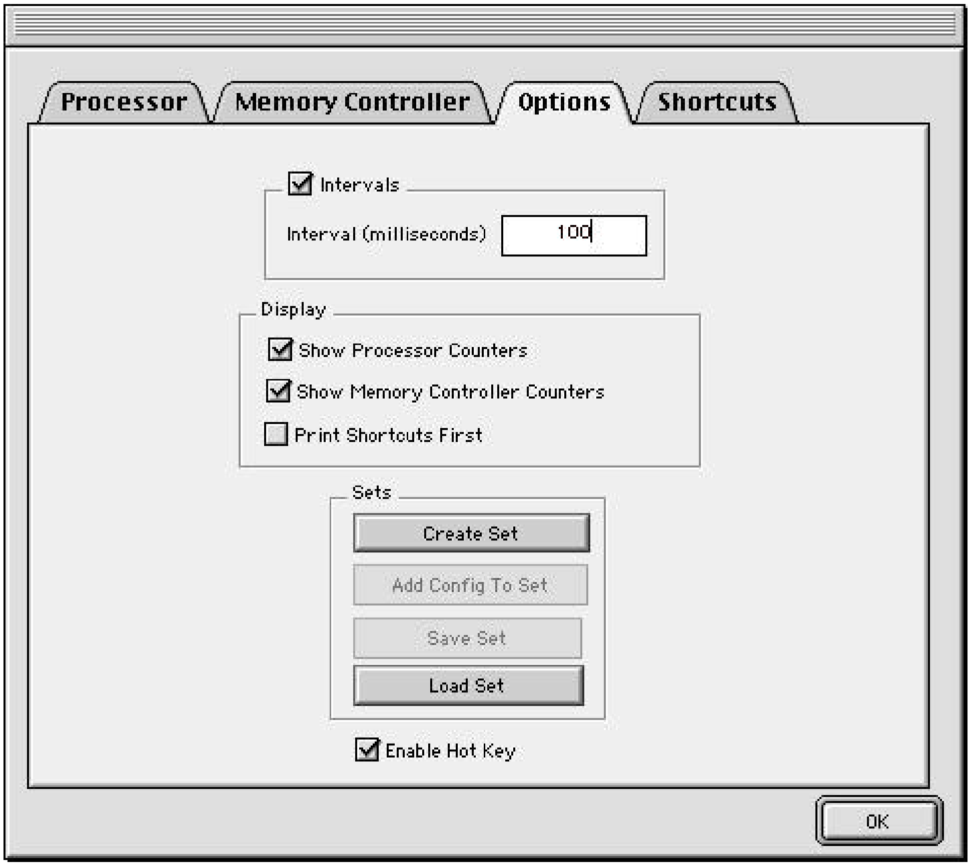


Figure 8. PCI Events Selection of Memory Controller Tab

Figure 9 shows the Options tab. Clicking the 'intervals' check box and entering a value in the text field configures 4PM to read the counters every N milliseconds where N is the value entered in the text field. When the check box is unchecked, 4PM will report a single value for each counter.

Three display options are given, 'Show Processor Counters', 'Show Memory Controller Counters', and 'Print Shortcuts First'. These refer to the output in the main text window of 4PM and consequently, the output written to a file when the measurement data is saved. The 'Enable Hot Key' check box enables the hot key. This means that 4PM installs its NMI routine The set buttons are covered in the section on configurations and sets below.



☑ Intervals

Interval (milliseconds) 100

Display

☑ Show Processor Counters

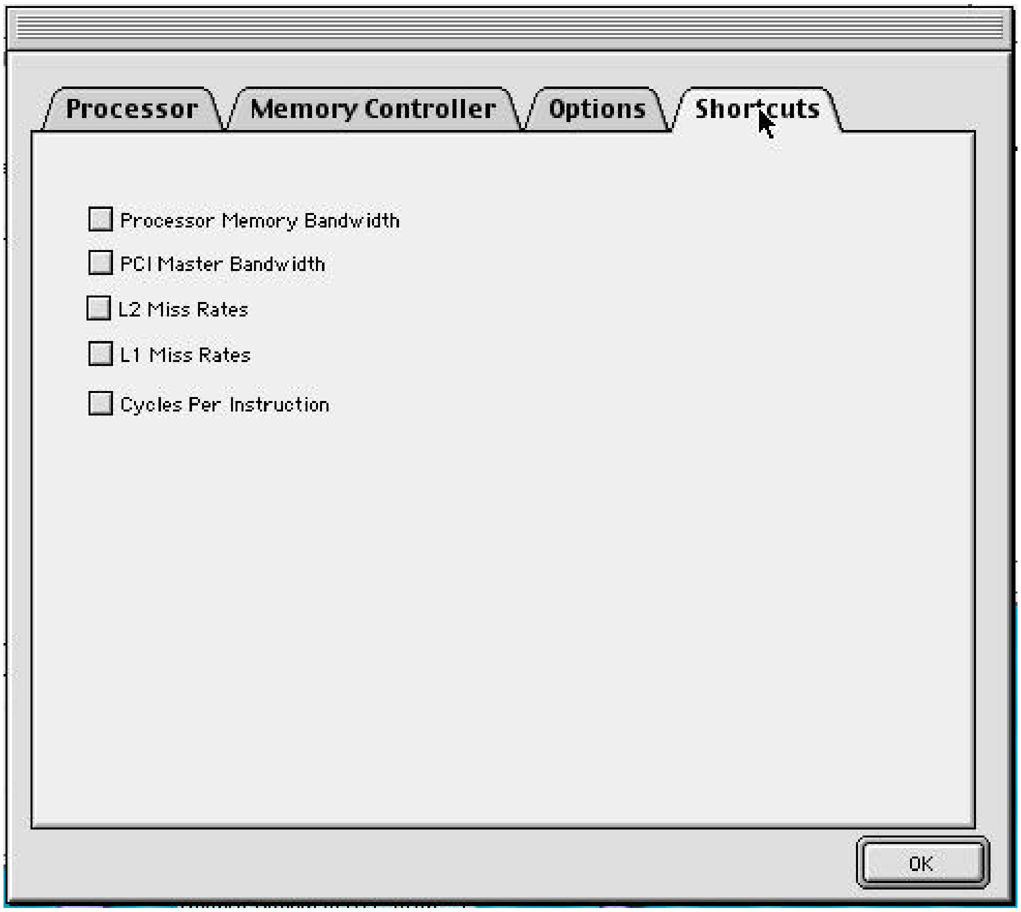
☑ Show Memory Controller Counters

□ Print Shortcuts First

☑ Enable Hot Key

Figure 9. Options Tab

Figure 10 shows the Shortcuts tab. A shortcut uses a combination of counters to report a higher level metric than an event count. Some of the shortcuts are not mutually exclusive meaning some of them may be recorded at the same time. When selecting a shortcut that can not be recorded at the same time as a previously selected shortcut (i.e. they need to use the same counters), the previously selected shortcut will be unselected (i.e. the box will be unchecked). Any selected shortcuts are shown in the 'shortcuts' box on 4PM's main window (shown in figure 1).



□ Processor Memory Bandwidth

□ PCI Master Bandwidth

□ L2 Miss Rates

□ L1 Miss Rates

□ Cycles Per Instruction

Figure 10. Shortcuts Tab

4PM allows saving and loading of configurations. A configuration is both the settings of all counters and the interval value (or lack of one). This makes it easy if there are frequently reused sets of counter values. The current 4PM configuration can be saved by selecting 'Save Configuration' from the 'File' menu. A configuration can be loaded using 'Load Configuration' from the 'File' menu. The 'File' menu is shown below.

The contents of the main scrolling window can be saved to a tab separated file by selecting 'Save' in the 'File' menu.

1. Starting and Stopping the Counters

4PM provides two ways to start and stop the counters. One way is to link the 4PM library to your projects and use start and stop calls directly from your source. This is explained in more detail in the section on the 4PM Library in this document. The other method to start and stop the counters is to use a 'hot key'. This is the *command* and *power on* keys pressed at the same time. Pressing these keys generates a NMI (non-maskable interrupt) which will be seen by 4PM. When 4PM sees a NMI, it will toggle the counters from on to off or from off to on.

1. Configurations and Sets

A configuration is defined as the event selected for each of the 8 counters (on current systems there will be 8 counters), the interval selection (off or some number of milliseconds) and the display options. These can be saved and reloaded using the *Load Configuration* and *Save Configuration* menu items in the *File* menu.

A set is a series of configurations that are run sequentially. After the final configuration in a set is run, 4PM will save the data from all of the runs. The same interval settings must be used for all configurations in a set. The set mechanism was added in attempt to facilitate automated measurements.

To create a set, click the *Create Set* button on the *Options* tab in the preferences dialog. This creates an empty set. Next, set the counters to the desired configuration and click the *Add Config To Set* button. This adds the configuration to the set. Continue changing counter events and adding configurations until you have all the configurations you want in the set. Click *Save Set* to save the set. You can name the set using the Nav Services dialog that is presented when you click *Save Set.*

To load and run a set, click on the *Load Set* button. Once the set is loaded, 4PM will run through the configurations in the set, advancing one configuration after the counters are stopped.

1. 4PM Library

The 4PM Library has four calls:

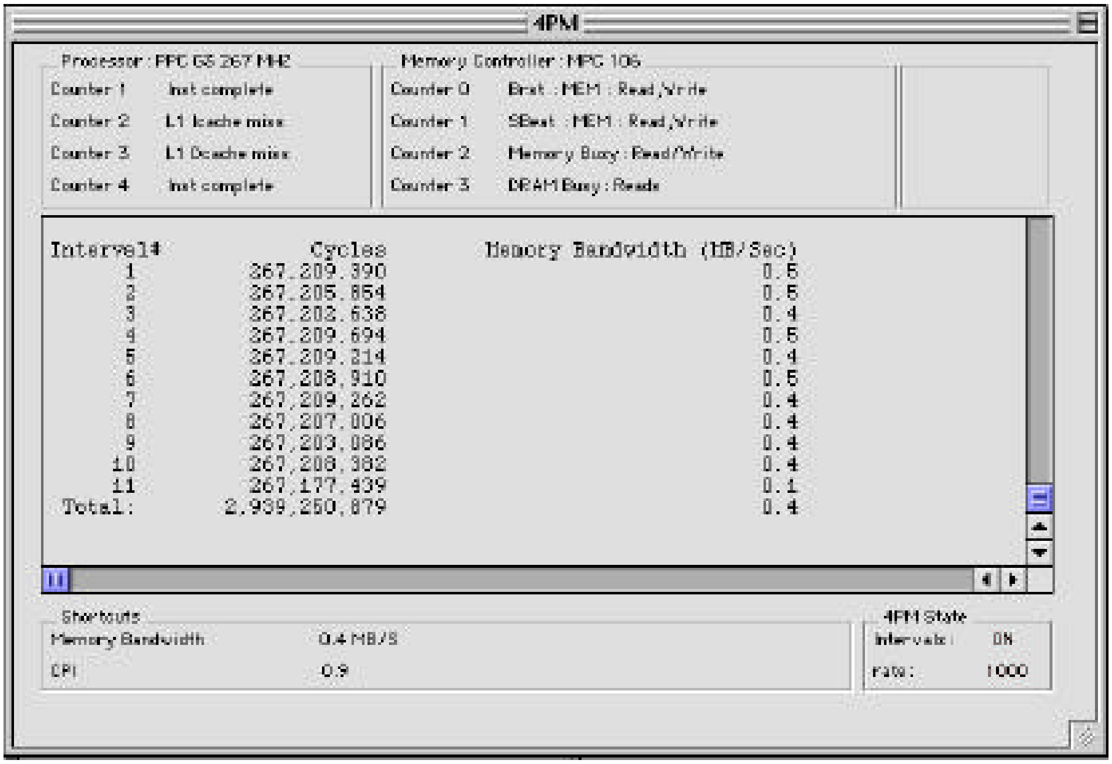
```
short   Init4PMLibrary(void);
void    Delete4PMLibrary(void);
void    Start4PM(void);
void    Stop4PM(void);
```

4PM must be running in the background while these calls are executed.

The Init4PMLibrary call will return an error code. It returns noErr (0) if initialization was successful, -1 if the machine is not running a supported processor (a PowerPC greater than 603) or PPCToolbox errors errors (see Inside Mac: Interapplication Communication).

Init4PMLibrary must be called exactly once before using the library and Delete4PMLibrary should be called before exiting the application that made the init call. Start4PM and Stop4PM may be called any number of times in start/stop sequence. Nesting of start calls is not supported.

Counts from multiple start/stop calls will continue accumulating until 4PM is brought to the front. If 4PM is configured to record on intervals, the interval data from multiple start/stop calls will be shown in one long stream of interval data with no notion of where the start and stop calls were.



Processor : PPC G3 267 MHz

Counter 1	Inst complete
Counter 2	L1 Icache miss
Counter 3	L1 Dcache miss
Counter 4	Inst complete

Memory Controller : MPC 106

Counter 0	Brst : MEM : Read,Write
Counter 1	SBeat : MEM : Read,Write
Counter 2	Memory Busy : Read/Write
Counter 3	DRAM Busy : Reads

```
Interval#           Cycles         Memory Bandwidth (MB/Sec)
      1        267,209,390                               0.5
      2        267,205,854                               0.5
      3        267,202,638                               0.4
      4        267,209,694                               0.5
      5        267,209,214                               0.4
      6        267,208,910                               0.5
      7        267,209,262                               0.4
      8        267,207,006                               0.4
      9        267,203,086                               0.4
     10        267,208,382                               0.4
     11        267,177,439                               0.1
 Total:      2,939,250,879                               0.4
```

Shortcuts

Memory Bandwidth	0.4 MB/S
CPI	0.9

4PM State

intervals :	ON
rate :	1000

Figure 11. Main Window with Interval Results

Figure 11 shows the main 4PM window displaying results from a test run. 4PM was configured to take readings at intervals of 1,000 milliseconds and to report the memory bandwidth and cpi. The bandwidth and cpi reported in the 'Shortcuts' box is the average over all recorded intervals. Scrolling the text to the right would show the cpi and the counts from the 8 counters.