Review: (Write on the board before class)
1) Computer designer: Functions, Last thru trends, HW vs. SW
2) Performance eq: CPU time = Instruction count * CPI * Clock cycle time

Today more performance, failure of the single measure
MIPS: most widely used, misleading figure

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} * 10^6} = \frac{\text{Clock rate}}{\text{CPI} * 10^6}$$

Bad (1) ISA dependent; (2) vary on machine; (3) inverse to perf (FP)
Example: 40% ALU, 1 clock, rest 2 clocks; opt. comp. discards 50% ALU (CPI 1.6 v. 1.75); 56 MHz clock; MIPS 35 v. 32; perf 1.6 v. 1.4

Relative MIPS as better than (native) MIPS

$$\text{Relative MIPS} = \frac{\text{Time}_{\text{reference}}}{\text{Time}_{\text{unrated}}} * \text{MIPS}_{\text{reference}}$$

Bad (1) which SW on reference machine? (2) still fix code & input (3) temptation to extrapolate; Good: (1) intuitive

MFLOPS: most widely used in scientific, also misleading

$$\text{MFLOPS} = \frac{\text{Number of floating-point operations in a program}}{\text{Execution time} * 10^6}$$

Bad (1) ops not constant (Cray vs. 68881) (2) MFLOPS vary (+ v. /) Normalized ops per program: + = 1, / = 4, SIN = 8 (3) for compiler? (4) tempation to generalize; Good: (1) intuitive

Must identify program and input, no FLOPS constant, so redundant to time & bad for integer codes

BENCHMARKS
(1) Programs; (2) kernels; (3) Toy; (4) Synthetic

Show Dhrystone & Whetstone

Fallacy bugs
(1) Compiler discards 25% of Dhry by removing loop overhead for code executed once, inline procedure expansion
(2) No Whet loops can be vectorized
(3) C byte copy => fixed double word transfer (20-30% improvement)
(4) Symbolic simplification of Whetstone

Class stuff:
Collect remaining homeworks (or in Evans Hall before class)

Collect Survey again
Questions?

Continue Lecture
Review: MIPS, MFLOPS, Benchmarks
Summarizing Performance

	Computer A	Computer B	Computer C
Program 1 (secs)	1	10	20
Program 2 (secs)	1000	100	20
Total time (secs)	1001	110	40

What can you say about relative performance?
Total time (workload = 1 program of each)

Same as arithmetic mean = $\frac{1}{n}\sum_{i=1}^{n}\text{Time}_i$

Calculate

But is this a good mix? Vary workload not even

weighted arithmetic mean = $\sum_{i=1}^{n}\text{Weight}_i * \text{Time}_i$,

Normalize? arithmetic mean of ratios != ratio of arithmetic means

To what? varies with machine used as reference (see Figure 2.8)
geometric mean is

$$\sqrt[n]{\prod_{i=1}^{n}\text{Execution Time Ratio}_i}$$

Same answer since raiot of means = means of ratios; not input dependent

$$\frac{\text{Geometric mean}(X_i)}{\text{Geometric mean}(Y_i)} = \text{Geometric mean}\left(\frac{X_i}{Y_i}\right)$$

No workload matches geometric mean!

Measure real workload, calcualte means; or equal time on some machine; first summarize, then ratios

Summarize: Job of computer designer: functions, trends, HW v. SW, perf eq.

Notes to lecturer: Stories on misleading MIPS, Find advertisements? (assign to students?).

```
      I = ITER
      ...
      N8 = 899 * I
      ...
      N11 = 93 * I
      ...
      X = 1.0
      Y = 1.0
      Z = 1.0
      IF (N8) 89,89,81
81    DO 88 I = 1, N8, 1
88       CALL P3(X,Y,Z)
89    CONTINUE
      ...
      X = 0.75
      IF (N11) 119,119,111
111   DO 118 I = 1, N11, 1
118      X = SQRT(EXP(ALOG(X)/T1))
119   CONTINUE
      ...
      SUBROUTINE P3 (X,Y,Z)
      COMMON T, TT1, T2
      X1 = X
      Y1 = Y
      X1 = T * (X1 + Y1)
      Y1 = T * (X1 + Y1)
      Z = (X1 + Y1) / T2
      RETURN
      END
      ...
```

FIGURE 2.4 Two loops of the Whetstone synthetic benchmark(optimize/vector P3?)

```
...
for(Run_Index = 1;
Run_Index<=Number_Of_Runs; ++Run_Index)
{
    Proc_5();
    Proc_4();
    Int_1_Loc = 2;
    Int_2_Loc = 3;
    strcpy(Str_2_Loc,"DHRYSTONE PROGRAMS,
2'ND STRING");
    ...
}
...
Proc_4()
{
    Boolean Bool_Loc;

    Bool_Loc = Ch1_1_Glob == 'A';
    Bool_Glob = Bool_Loc | Bool_Glob;
    Ch1_2_Glob = 'B';
} /* Proc_4 */

Proc_5()
{
    Ch1_1_Glob = 'A';
    Bool_Glob = false;
} /* Proc_5 */
...
```

FIGURE 2.5 A section of the Dhrystone synthetic benchmark. (inline P5(), string copy)

	A	B	C	W(1)	W(2)	W(3)
Program 1 (secs)	1.00	10.00	20.00	0.50	0.909	0.999
Program 2 (secs)	1000.00	100.00	20.00	0.50	0.091	0.001
Arithmetic mean :W(1)	500.50	55.00	20.00			
Arithmetic mean :W(2)	91.82	18.18	20.00			
Arithmetic mean :W(3)	2.00	10.09	20.00			

FIGURE 2.7 Weighted arithmetic mean execution times using three weightings. (W1: =mix, W2: =time B, W3: =timeA)

	Normalized to Comp. A (%)			**Normalized to Comp. B (%)**			**Normalized to Comp. C (%)**		
	A	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**
Program 1	100	1,000	2,000	10	100	200	5	50	100
Program 2	100	10	2	1000	100	20	5,000	500	100
Arithmetic mean	100	505	1,001	505	100	110	2,503	275	100
Geometric mean	100	100	63	100	100	63	158	158	100
Total time	100	11	4	910	100	36	2,503	275	100

FIGURE 2.8 Execution times from Figure 2.6 normalized to each machine (in %). (bigger=slower; A= 5XB vs B = 5XA)