

Misura delle prestazioni

E' fondamentale definire una metrica corretta per misurare le prestazioni di un sistema di elaborazione.

- Prospettiva dell'acquirente

- dato un insieme di macchine, quale ha

- le migliori prestazioni ?
- il minor costo ?
- Il miglior rapporto prestazioni/costo ?

- Prospettiva del progettista

- di fronte a possibili soluzioni progettuali alternative, quale presenta

- il miglior incremento delle prestazioni ?
- il minor costo ?
- il miglior rapporto prestazioni/costo ?

Come si misurano le prestazioni ?

Quali sono le grandezze tipicamente usate ?

frequenza di clock

dimensione della RAM

dimensione dei dischi

Qual è il parametro che ci interessa realmente ?

Due concetti di prestazioni

	DC to Paris	Speed	Passengers	Throughput (pmp)
Boeing 747	6.5 hours	610 mph	470	286,700
BAC/Sud Concorde	3 hours	1350 mph	132	178,200

Quale aereo ha le migliori prestazioni ?

- ° Prospettiva del passeggero

Tempo impiegato per un singolo viaggio

- ° Prospettiva della compagnia aerea

Numero di passeggeri trasferiti in un dato intervallo di tempo

Due misure possibili

- **Tempo di risposta (o di esecuzione)**
 - Quanto tempo devo aspettare per l'uscita del mio programma?
 - Quanto tempo per lanciare il mio programma ?
 - Quanto tempo per eseguire il mio programma ?
- **Throughput**
 - Quanto lavoro viene svolto?
 - Quanti programmi possono essere eseguiti insieme?
 - Qual è il rate di esecuzione medio ?

Quali sono gli effetti dovuti a:

- *upgrade di un computer con un processore più veloce*
- *acquisto di un nuovo computer*

Prestazione = f(tempo di esecuzione)

- **Per un dato programma eseguito sulla macchina X,**

$$\text{Prestazione}(X) = 1 / \text{tempo di esecuzione}(X)$$

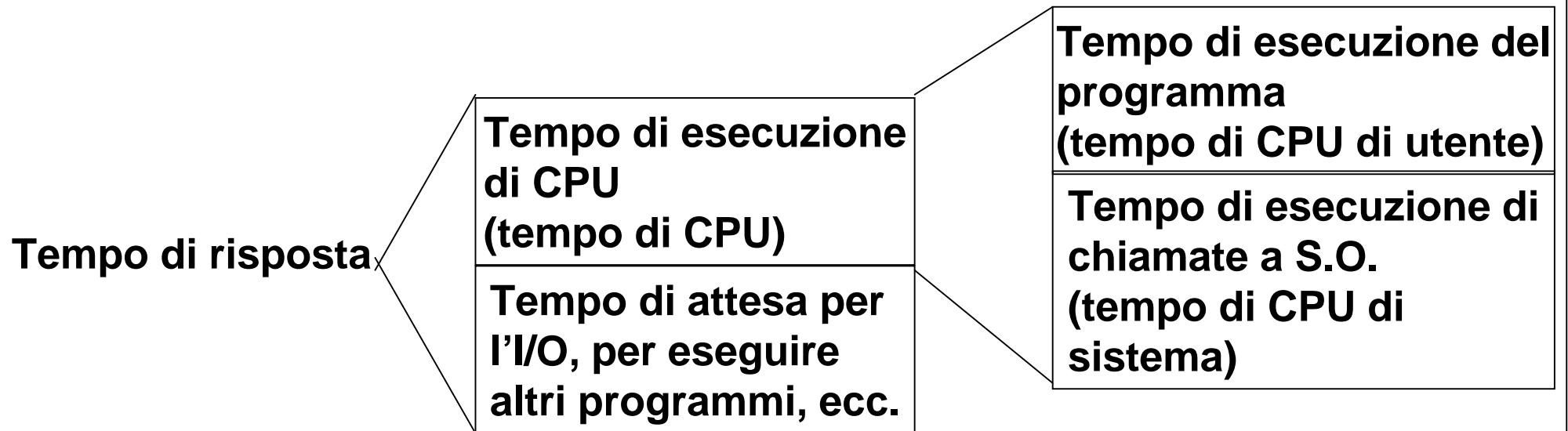
- **“X è n volte più veloce di Y” se:**

$$\text{Prestazione}(X) / \text{Prestazione}(Y) = n$$

- **Problema:**

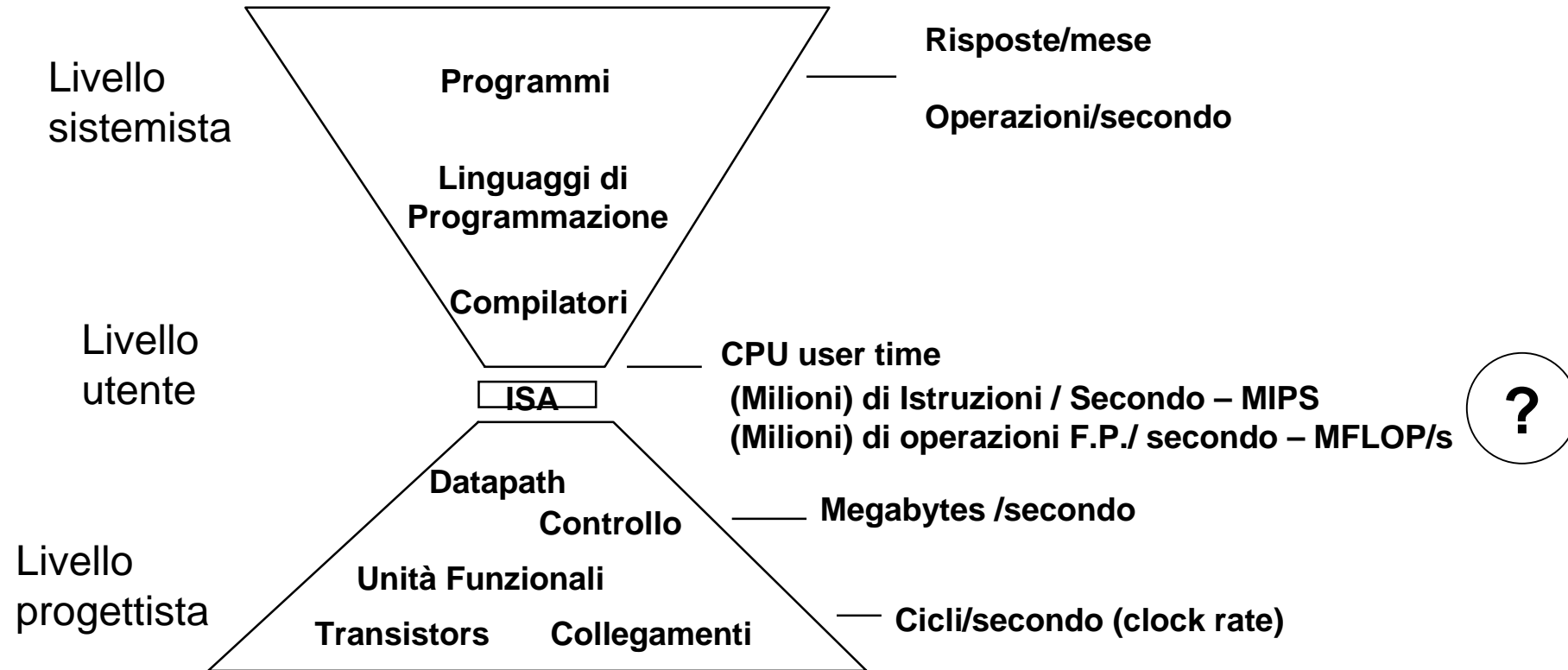
- **la macchina A esegue un programma in 20 secondi**
- **la macchina B esegue lo stesso programma in 25 secondi**

Che tempo fa ?



Parametro di riferimento: tempo di CPU di utente (*user CPU time*)

Metriche per misurare le prestazioni



Diverse metriche per diversi aspetti (e per diversi abusi)
Come si relazionano le diverse metriche ?

Programma sorgente (HLL)

Compilatore

Programma in Linguaggio Assembly

Assembler

Programma in Linguaggio Macchina

Interpretazione della macchina

Definizione dei segnali di controllo

- o
- o

```
temp = v[k];  
v[k] = v[k+1];  
v[k+1] = temp;
```

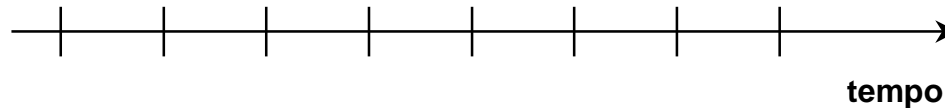
```
lw $15, 0($2)  
lw $16, 4($2)  
sw $16, 0($2)  
sw $15, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000  
1010 1111 0101 1000 0000 1001 1100 0110  
1100 0110 1010 1111 0101 1000 0000 1001  
0101 1000 0000 1001 1100 0110 1010 1111
```

```
ALUOP[0:3] <= InstReg[9:11] & MASK
```


Clock

- **Ogni calcolatore è sincronizzato da un orologio interno che procede a velocità costante (clock)**
- **I “clock ticks” definiscono gli istanti possibili per la realizzazione di eventi in hardware (evoluzione dello stato della macchina) :**



- **tempo di ciclo= intervallo tra due ticks = secondi per ciclo**
- **clock rate (frequenza) = cicli al secondo (1 Hz. = 1 ciclo/sec)**

Un clock da 500 Mhz ha un tempo di ciclo

$$\frac{1}{500 \times 10^6} \times 10^9 = 2 \text{ nanosecond i}$$

Come aumentare le prestazioni

E' possibile esprimere il tempo di esecuzione in termini di cicli di clock

$$\frac{\text{secondi}}{\text{programma}} = \frac{\text{cicli}}{\text{programma}} \times \frac{\text{secondi}}{\text{ciclo}}$$

A parità di altre condizioni, si ottiene un aumento delle prestazioni

se

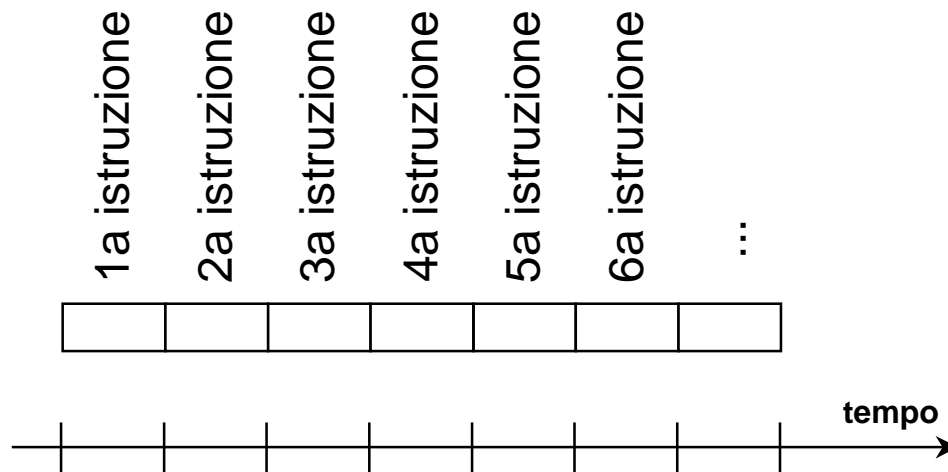
_____ **il numero di cicli/programma, oppure**

_____ **il tempo di ciclo del clock o, equivalentemente,**

_____ **il clock rate.**

Quanti cicli per eseguire un programma ?

- Si può assumere numero di cicli = numero di istruzioni ?

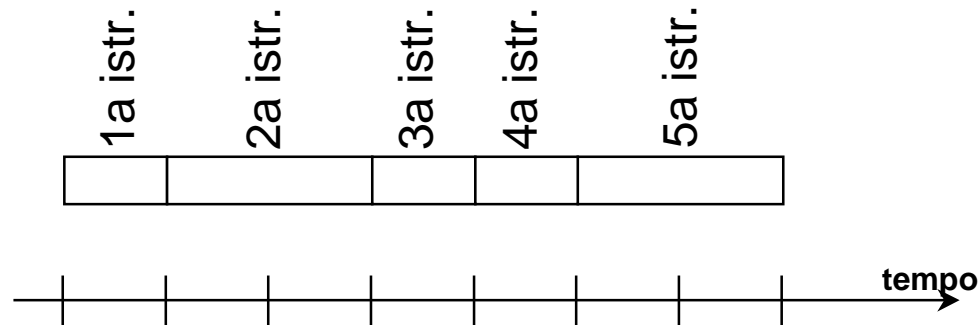


No !

Istruzioni differenti comportano tempi diversi su macchine differenti.

Perché ?

Istruzioni differenti richiedono numeri differenti di cicli



- **Un'operazione di moltiplicazione richiede un tempo maggiore rispetto ad un'addizione**
- **Un'operazione tra floating point richiede un tempo maggiore rispetto ad una tra interi**
- **Un accesso in memoria richiede un tempo maggiore rispetto ad un accesso a registri interni**
- ***Achtung !!! - una modifica del tempo di ciclo comporta spesso una modifica del numero di cicli richiesti per le istruzioni***

Esempio

Un programma viene eseguito in 10 secondi sul computer A, che ha un clock da 400 Mhz. E' però necessario che venga eseguito in 6 secondi e per questo motivo si intende costruire una nuova macchina B con una nuova tecnologia di realizzazione della CPU che permette un incremento notevole della frequenza di clock, ma a spese del numero di cicli per istruzione che porterebbe ad un aumento del 20% del numero di cicli richiesti per l'esecuzione del programma.

Quale frequenza di clock deve assicurare la macchina B ?

Equazione fondamentale:

$$\frac{\text{secondi}}{\text{programma}} = \frac{\text{cicli}}{\text{programma}} \times \frac{\text{secondi}}{\text{ciclo}}$$

Dove sono le istruzioni ?

$$\frac{\text{secondi}}{\text{programma}} = \frac{\text{cicli}}{\text{programma}} \times \frac{\text{secondi}}{\text{ciclo}}$$

Nell'equazione considerata non c'è riferimento (esplicito) al numero di istruzioni che formano il programma; chiaramente, questo è un parametro che influisce sul tempo di esecuzione.

Come si può esplicitare il rapporto con il numero di istruzioni ?

Definiamo il numero medio di cicli per istruzione: CPI (*clock cycles per instruction*).

$$\text{CPI} = \frac{\text{numero cicli di clock del programma}}{\text{numero istruzioni del programma}}$$

$$\frac{\text{secondi}}{\text{programma}} = \text{numero istruzioni} \times \text{CPI} \times \frac{\text{secondi}}{\text{ciclo}}$$

Chi è responsabile ?

$$\text{Cpu time} = \frac{\text{secondi}}{\text{programma}} = \text{numero istruzioni} \times \text{CPI} \times \frac{\text{secondi}}{\text{ciclo}}$$

	num. istr.	CPI	clock rate
Programma			
Compilatore			
ISA			
Organizzaz.			
Tecnologia			

Chi è responsabile ?

$$\text{Cpu time} = \frac{\text{secondi}}{\text{programma}} = \text{numero istruzioni} \times \text{CPI} \times \frac{\text{secondi}}{\text{ciclo}}$$

	num. istr.	CPI	clock rate
Programma	X		
Compilatore	X	X	
ISA	X	X	X
Organizzaz.		X	X
Tecnologia			X

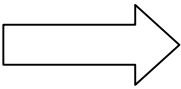
Istruzioni e CPI

Le istruzioni che contribuiscono a formare il parametro CPI sono di diverso tipo e caratterizzate da differenti numeri di cicli per l'esecuzione.

E' quindi possibile raggruppare le diverse istruzioni in classi caratterizzate dallo stesso numero di cicli di clock, rendendo esplicito il contributo di ciascuna classe al valore di CPI

$$\text{cicli totali} = \sum_{i=1}^n \text{CPI}_i \times n_i$$

CPI_i = CPI per le istruzioni della classe i-ma f_i = frequenza della classe i-ma
 n_i = numero di istruzioni della classe i-ma N = numero totale istruzioni


$$\text{CPI} = \frac{\text{cicli totali}}{N} = \frac{\sum_{i=1}^n \text{CPI}_i \times n_i}{N} = \sum_{i=1}^n \text{CPI}_i \times \frac{n_i}{N} = \sum_{i=1}^n \text{CPI}_i \times f_i$$

In quale direzione andare per migliorare il CPI ?

Esempio

Una macchina è caratterizzata dalla seguente distribuzione di CPI su tre classi

Classe	CPI
A	1
B	2
C	3

Una particolare istruzione in HLL può essere tradotta da un compilatore tramite due sequenze possibili, che usano combinazioni diverse di istruzioni delle tre classi, secondo la tabella seguente:

Sequenza	A	B	C
1	2	1	2
2	4	1	1

Qual è la soluzione migliore ?

Esempio (processore RISC)

Base Machine (Reg / Reg)

Op	f_i	CPI_i	$f_i \times CPI_i$	% Tempo
ALU	50%	1	.5	23%
Load	20%	5	1.0	45%
Store	10%	3	.3	14%
Branch	20%	2	.4	18%
			<hr/>	
			2.2	

Mix tipico

Di quanto sarebbe più veloce la macchina se si usasse una cache che portasse a 2 il numero di cicli necessari per un load ?

E se si impiegasse una tecnica di branch prediction per risparmiare un ciclo dei due richiesti per un branch ?

Che cosa succederebbe se si potessero eseguire due istruzioni ALU contemporaneamente ?

La legge di Amdahl

Il miglioramento di prestazione (accelerazione o *speedup*) ottenibile mediante l'uso di alcune modalità di esecuzione più veloci è limitato dalla frazione di tempo in cui queste modalità possono essere impiegate.

Supponiamo di aver apportato una modifica E ad una certa macchina.

Lo speedup che si ottiene si valuta come:

$$\text{Speedup}(E) = \frac{\text{Prestazione (dopo di E)}}{\text{Prestazione (prima di E)}} = \frac{\text{ExTime (prima di E)}}{\text{ExTime (dopo di E)}}$$

Supponiamo che la modifica E porti ad un'accelerazione pari ad un fattore S di una frazione F dell'intero task, mentre il resto rimane inalterato:

$$\text{ExTime(dopo di E)} = ((1-F) + F/S) \times \text{ExTime(prima di E)}$$

$$\text{Speedup(dopo di E)} = \frac{1}{(1-F) + F/S}$$

Programmi per valutare le prestazioni

Quale programma usare per valutare le prestazioni di una macchina ?

L'ideale sarebbe di usare l'insieme di programmi che si sa di dover eseguire sulla macchina (*workload*), ma non è sempre possibile.

In alternativa, si potrebbero usare dei programmi campione (*benchmark*).

Difficoltà:

- le caratteristiche del benchmark devono essere simili a quelle del workload
- il benchmark deve essere standard

Soluzioni:

- benchmark sintetici (Whetstone, Dhrystone, kernel benchmark)
- mix di applicazioni reali (SPEC)

SPEC (*System Performance Evaluation Cooperative*)

Open Systems Group (OSG):

SPEC CPU2000: benchmark per le prestazioni di CPU

SPEC JBB2000: benchmark Java server-side.

SPEC JVM98: benchmark Java Virtual Machine.

SPEC MAIL2001: benchmark mail-server.

SPEC WEB99: benchmark per server WWW.

SPEC SDM91: benchmark per Multi-user UN*X commands.

SPEC SFS97_R1 (3.0): benchmark per File Server.

Graphics Performance Characterization Group (GPC):

GLperf Results: benchmark per operazioni grafiche OpenGL 2D and 3D. graphics operations (points, lines, triangles, pixels, etc.)

SPECapc for Pro/ENGINEER Results: Standardized graphics performance tests.

High Performance Group (HPG):

SPEC HPC96 Results: the SPEC "supercomputer" tests.

SPEC OMP2001 Results: the SPEC OpenMP/SMP tests.

SPEC CPU2000

E' una suite di benchmark studiati per misurare le prestazioni rispetto ad applicazioni di forte peso computazionale. Vengono quindi evidenziati le prestazioni di:

- CPU
- architettura di memoria
- compilatori

Vengono valutate distintamente le prestazioni su operazioni di tipo intero e floating point:

- SPECint2000, SPECfp2000
misurano la velocità di esecuzione
- SPECint_rate2000, SPECfp_rate2000
misurano il throughput.

SPECint 2000

<i>Nome</i>	<i>Ref. Time</i>	<i>Applicazione</i>
164.gzip	1400	Data compression utility
175.vpr	1400	FPGA circuit placement and routing
176.gcc	1100	C compiler
181.mcf	1800	Minimum cost network flow solver
186.crafty	1000	Chess program
197.parser	1800	Natural language processing
252.eon	1300	Ray tracing
253.perlbnk	1800	Perl
254.gap	1100	Computational group theory
255.vortex	1900	Object Oriented Database
256.bzip2	1500	Data compression utility
300.twolf	3000	Place and route simulator

SPECfp 2000

<i>Nome</i>	<i>Ref. Time</i>	<i>Applicazione</i>
168.wupwise	1600	Quantum chromodynamics
171.swim	3100	Shallow water modeling
172.mgrid	1800	Multi-grid solver in 3D potential field
173.applu	2100	Parabolic/elliptic partial differential equations
177.mesa	1400	3D Graphics library
178.galgel	2900	Fluid dynamics: analysis of oscillatory instability
179.art	2600	Neural network simulation; adaptive resonance theory
183.equake	1300	Finite element simulation; earthquake modeling
187.facerec	1900	Computer vision: recognizes faces
188.ammp	2200	Computational chemistry
189.lucas	2000	Number theory: primality testing
191.fma3d	2100	Finite element crash simulation
200.sixtrack	1100	Particle accelerator model
301.apsi	2600	Solves problems regarding temperature, wind, velocity and distribution of pollutants

CINT2000 (for integer compute intensive performance comparisons):

- SPECint2000: The geometric mean of twelve normalized ratios (one for each integer benchmark) when compiled with aggressive optimization for each benchmark.
- SPECint_base2000: The geometric mean of twelve normalized ratios when compiled with conservative optimization for each benchmark.
- SPECint_rate2000: The geometric mean of twelve normalized throughput ratios when compiled with aggressive optimization for each benchmark.
- SPECint_rate_base2000: The geometric mean of twelve normalized throughput ratios when compiled with conservative optimization for each benchmark.



CINT2000 Result

Copyright ©1999-2001, Standard Performance Evaluation Corporation

Intel Corporation

Intel D850GB motherboard (1.8 GHz, Pentium 4 processor)

SPECint2000 = 613

SPECint_base2000 = 598

SPEC license #: 13 Tested by: Intel Corporation Test date: Aug-2001 Hardware Avail: Jul-2001 Software Avail: Jun-2001

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio	
164.gzip	1400	214	654	214	655	
175.vpr	1400	429	327	404	347	
176.gcc	1100	172	640	172	640	
181.mcf	1800	357	504	357	504	
186.crafty	1000	172	582	171	584	
197.parser	1800	325	553	325	554	
252.eon	1300	158	822	135	961	
253.perlbmk	1800	214	841	214	840	
254.gap	1100	133	825	131	838	
255.vortex	1900	234	810	224	849	
256.bzip2	1500	307	488	304	493	
300.twolf	3000	714	420	713	421	

Hardware

CPU: Pentium 4 processor (1.8 GHz, 400 MHz bus)
CPU MHz: 1800
FPU: Integrated
CPU(s) enabled: 1
CPU(s) orderable: 1
Parallel: No
Primary Cache: 12k micro-ops I + 8KBD on chip
Secondary Cache: 256KB(I+D) on chip
L3 Cache: N/A
Other Cache: N/A
Memory: 256 MB (2 128 MB PC800 RDRAM non-ECC modules)
Disk Subsystem: IBM DTLA-307030 ATA-100
Other Hardware: None

Software

Operating System: Windows 2000(Build 2195, SP2)
Compiler: Intel C++ Compiler 5.0.1 Build 010525Z
Microsoft Visual C++ 6.0 SP5(for libraries)
MicroQuill SmartHeap library V6.0
File System: FAT32
System State: Default

Notes/Tuning Information

```
+FDO: PASS1=-Qprof_gen PASS2=-Qprof_use
Base tuning for C programs: -QxW -Qipo shlw32M.lib +FDO
Base tuning for C++ programs: -QxW -Qipo -GX -GR
shlw32M.lib is the SmartHeap library V6.0 from MicroQuill www.microquill.com
Portability flags:
176.gcc: -Dalloca=_alloca /F10000000
186.crafty: -DNT_i386
253.perlbmk: -DSPEC_CPU2000_NTOS -DPERLDLL /MT
254.gap: -DSYS_HAS_CALLOC_PROTO -DSYS_HAS_MALLOC_PROTO
Peak tuning:
164.gzip: -QxW -Qipo -Oa shlw32M.lib +FDO
175.vpr: -QxW -Qwp_ipo +FDO
176.gcc: -QxW -Qipo shlw32M.lib +FDO
181.mcf: -QxW -Qipo shlw32M.lib +FDO
186.crafty: -QxW -Qipo -Oa shlw32M.lib +FDO
197.parser: -QxW -Qipo shlw32M.lib +FDO
252.eon: -QxW -Qipo +FDO
253.perlbmk: -QxW -Qipo shlw32M.lib +FDO
254.gap: -QxW -Qipo +FDO
255.vortex -QxW -Qwp_ipo -Oa shlw32M.lib +FDO
```



CINT2000 Result

Copyright ©1999-2001, Standard Performance Evaluation Corporation

Intel Corporation

Intel D850GB motherboard (1.8 GHz, Pentium 4 processor)

SPECint2000 = 613

SPECint_base2000 = 598

SPEC license #: 13 | Tested by: Intel Corporation | Test date: Aug-2001 | Hardware Avail: Jul-2001 | Software Avail: Jun-2001

Notes/Tuning Information (Continued)

256.bzip2: -QxW -Qipo -Oa +FDO

300.twolf: -QxW -Qipo shlw32M.lib +FDO

Tested systems can be used with Shin-G ATX case, Delta Inc. power supply PS-300GB-1 B

Product description located as of 11/2000:

http://developer.intel.com/design/motherbd/gb/gb_ds.htm

Motherboard is available through Intel OEM Distribution Channel. See

<http://channel.intel.com/business/ibp/boards/d850gb.htm> for more information

The system bus runs at 400 MHz



CFP2000 Result

Copyright ©1999-2001, Standard Performance Evaluation Corporation

Intel Corporation

Intel D850GB motherboard(1.8 GHz, Pentium 4 processor)

SPECfp2000 = 628

SPECfp_base2000 = 618

SPEC license #: 13 Tested by: Intel Corporation Test date: Apr-2001 Hardware Avail: Jul-2001 Software Avail: Oct-2000

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio	
168.wupwise	1600	179	894	179	894	
171.swim	3100	241	1285	241	1285	
172.mgrid	1800	283	637	263	685	
173.applu	2100	285	736	285	738	
177.mesa	1400	209	670	214	655	
178.galgel	2900	485	598	485	598	
179.art	2600	495	525	482	539	
183.quake	1300	160	812	142	916	
187.facerec	1900	360	528	358	531	
188.ammmp	2200	554	397	550	400	
189.lucas	2000	232	862	232	862	
191.fma3d	2100	425	495	425	494	
200.sixtrack	1100	359	306	355	310	
301.apsi	2600	551	472	551	472	

Hardware

CPU: Pentium 4 processor (1.8 GHz, 400 MHz bus)
CPU MHz: 1800
FPU: Integrated
CPU(s) enabled: 1
CPU(s) orderable: 1
Parallel: No
Primary Cache: 12k micro-ops I + 8KBD
Secondary Cache: 256KB(I+D) on chip
L3 Cache: N/A
Other Cache: N/A
Memory: 256 MB (2 128 MB PC800 RDRAM non-ECC modules)
Disk Subsystem: IBM DTLA-307030 ATA-100
Other Hardware: None

Software

Operating System: Windows 2000 (Build 2195) Service Pack 1
Compiler: Intel C/C++ Compiler 5.0 Build 001120,
Intel Fortran Compiler 5.0 Build 001027,
Microsoft Visual C++ 6.0(for libraries)
Other software: Microquill SmartHeap V5.0
File System: FAT32
System State: Default

Notes/Tuning Information

```
+FDO: PASS1= -Qprof_gen PASS2=-Qprof_use
Base tuning: -Qipo -QxW -O3 +FDO
Portability:
178.galgel: -FI /F32000000
Peak tuning:
168.wupwise: -Qipo -QxW -O3 +FDO
171.swim: -Qipo -QxW -O3 +FDO
172.mgrid: -Qipo -QaxW -O3 +FDO
173.applu: -Qipo -QxW -O3 +FDO
177.mesa: -Qipo -QxW -Oa -O3 +FDO
178.galgel: -Qipo -QxW -O3 +FDO
179.art: -Qipo -QxW -O3 -Oa +FDO shlw32M.lib
183.quake: -Qipo -QxW -Qrcd -Oa +FDO shlw32M.lib
187.facerec: -Qipo -QxW -O3 +FDO shlw32M.lib
188.ammmp: -Qipo -Oa -QxW -O3 +FDO
189.lucas: -Qipo -QxW -O3 +FDO
191.fma3d: -Qipo -QxW -O3 +FDO
```



CFP2000 Result

Copyright ©1999-2001, Standard Performance Evaluation Corporation

Intel Corporation

Intel D850GB motherboard(1.8 GHz, Pentium 4 processor)

SPECfp2000 = 628

SPECfp_base2000 = 618

SPEC license #: 13 | Tested by: Intel Corporation | Test date: Apr-2001 | Hardware Avail: Jul-2001 | Software Avail: Oct-2000

Notes/Tuning Information (Continued)

200.sixtrack: -Qipo -QxW +FDO
 301.apsi: -Qipo -QxW -O3 +FDO
 Library ordering for 187.facerec (include SmartHeap correctly with default libs):
 LIBS=libIEPCF90.lib libintrins.lib libF90.lib libqwind.lib libm.lib shlW32M.lib
 LIBC.lib libirc.lib OLDNAMES.lib
 shlW32M.lib is the SmartHeap library V5.0 from Microquill: <http://www.microquill.com>
 Tested systems can be used with Shin-G ATX case, Delta Inc. power supply PS-300GB-1 B
 Product description located as of 11/2000:
http://developer.intel.com/design/motherbd/gb/gb_ds.htm
 Motherboard is available through Intel OEM Distribution Channel. See
<http://channel.intel.com/business/ibp/boards/d850gb.htm> for more information
 The system bus runs at 400 MHz