Homework 1 Solution

2019年10月13日 星期日　上午9:57

1.2

a. Performance via Pipelining
b. Dependability via Redundancy
c. Performance via Prediction
d. Make the Common Case Fast
e. Hierarchy of Memories
f. Performance via Parallelism
g. Design for Moore's Law
h. Use Abstraction to Simplify Design

1.3 The program is compiled into an assembly language program, which is then assembled into a machine language program.

1.5

a. performance of P1 (instructions/sec) $= 3 \times 10^9/1.5 = 2 \times 10^9$
performance of P2 (instructions/sec) $= 2.5 \times 10^9/1.0 = 2.5 \times 10^9$
performance of P3 (instructions/sec) $= 4 \times 10^9/2.2 = 1.8 \times 10^9$

b. cycles(P1) $= 10 \times 3 \times 10^9 = 30 \times 10^9$ s
cycles(P2) $= 10 \times 2.5 \times 10^9 = 25 \times 10^9$ s
cycles(P3) $= 10 \times 4 \times 10^9 = 40 \times 10^9$ s

c. No. instructions(P1) $= 30 \times 10^9/1.5 = 20 \times 10^9$
No. instructions(P2) $= 25 \times 10^9/1 = 25 \times 10^9$
No. instructions(P3) $= 40 \times 10^9/2.2 = 18.18 \times 10^9$
$CPI_{new} = CPI_{old} \times 1.2$, then CPI(P1) = 1.8, CPI(P2) = 1.2, CPI(P3) = 2.6
f = No. instr. $\times$ CPI/time, then

f(P1) $= 20 \times 10^9 \times 1.8/7 = 5.14$ GHz
f(P2) $= 25 \times 10^9 \times 1.2/7 = 4.28$ GHz
f(P1) $= 18.18 \times 10^9 \times 2.6/7 = 6.75$ GHz

1.6

a. Class A: 10^5 instr. Class B: 2×10^5 instr. Class C: 5×10^5 instr. Class D: 2×10^5 instr.

Time = No. instr. $\times$ CPI/clock rate

Total time P1 $= (10^5 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 3 + 2 \times 10^5 \times 3)/(2.5 \times 10^9) = 10.4 \times 10^{-4}$ s

Total time P2 $= (10^5 \times 2 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 2 + 2 \times 10^5 \times 2)/(3 \times 10^9) = 6.66 \times 10^{-4}$ s

$CPI(P1) = 10.4 \times 10^{-4} \times 2.5 \times 10^9/10^6 = 2.6$

$CPI(P2) = 6.66 \times 10^{-4} \times 3 \times 10^9/10^6 = 2.0$

b. clock cycles(P1) $= 10^5 \times 1 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 3 + 2 \times 10^5 \times 3 = 26 \times 10^5$

clock cycles(P2) $= 10^5 \times 2 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 2 + 2 \times 10^5 \times 2 = 20 \times 10^5$

Exercise 1.9.1. (# Processors affects Execution Time)

Here's a table with all the calculations (you didn't have to go into all this detail, but it would have helped for partial credit). We had a clock rate of 2 GHz.

p	N_A	CPI_A	C_A	N_L	CPI_L	C_L	N_B	CPI_B	C_B	*C*	*Exec*	*Speedup*
1	2.56	1	2.56	1.28	12	15.36	0.256	5	1.28	19.20	9.6	1.00
2	1.83	1	1.83	0.91	12	10.97	0.256	5	1.28	14.08	7.04	1.36
4	0.91	1	0.91	0.46	12	5.49	0.256	5	1.28	7.68	3.84	2.50
8	0.46	1	0.46	0.23	12	2.74	0.256	5	1.28	4.48	2.24	4.29

Key: (The $N_{...}$ and $C_{...}$ columns are in units of 10^9)

- p = number of processors
- A, L, B are the instruction categories (Arith, Load/Store, and Branch)
- N_A, N_L, N_B are the number of instructions by category
  - We're given $N_A = 2.56/(0.7\,p)$, $N_L = 1.28/(0.7\,p)$, and $N_B = 0.256$.
- CPI_A, CPI_L, CPI_B are the CPI rates, by category (given as 1, 12, and 5)
- C_A, C_L, C_B are the number of clock cycles by category. ($C_A = N_A \times CPI_A$, etc.)
- $C = C_A + C_L + C_B$ is the total number of clock cycles.
- *Exec* is execution time in seconds = $C/2$ GHz clock rate
- *Speedup* is execution time relative to 1 processor

1.9.2

$new\ C = old\ C + (new\ CA - old\ CA)$

1 Processor (slowdown 1.13)

9.6 + (1 × 2.56 × 10^9) / 2GHz = 10.88 s

2 Processors (slowdown 1.13)

(7.04 + (1 × 2.56 × 10^9)) / (0.7 × 2 × 2GHz) = 7.95 s

4 Processors (slowdown 1.12)

(3.84 + (1 × 2.56 × 10^9)) / (0.7 × 4 × 2GHz) = 4.3 s

8 Processors (slowdown 1.1)

(7.04 + (1 × 2.56 × 10^9)) / (0.7 × 8 × 2GHz) = 2.47 s

Exercise 1.9.3. (Change CPI of Load/Store to reduce Execution Time)

To decrease the execution time for 1 processor ($C = 19.20$ G cycles) to match the execution time for 4 processors ($C = 7.68$), we need to lower C_L from 15.36 to $(15.36 - (19.20 = 7.68)) = 15.36 - 11.52 = 3.84$. So new CPI_L = (new C_L) / N_L = 3.84 / 1.28 = 3.0.

1.11

1.11.1 CPI = clock rate × CPU time/instr. count

clock rate = 1/cycle time = 3 GHz

$CPI(bzip2) = 3 \times 10^9 \times 750/(2389 \times 10^9) = 0.94$

1.11.2 SPEC ratio = ref. time/execution time

SPEC ratio(bzip2) = 9650/750 = 12.86

1.11.3. CPU time = No. instr. × CPI/clock rate

If CPI and clock rate do not change, the CPU time increase is equal to the increase in the of number of instructions, that is 10%.

1.11.4 CPU time(before) = No. instr. × CPI/clock rate

CPU time(after) = 1.1 × No. instr. × 1.05 × CPI/clock rate

CPU time(after)/CPU time(before) = 1.1 × 1.05 =1.155. Thus, CPU time is increased by 15.5%.

1.11.5 SPECratio = reference time/CPU time

SPECratio(after)/SPECratio(before) = CPU time(before)/CPU time(after) = 1/1.1555 = 0.86. The SPECratio is decreased by 14%.

1.12

1.12.1 $T(P1) = 5 \times 10^9 \times 0.9 / (4 \times 10^9) = 1.125$ s

$T(P2) = 10^9 \times 0.75 / (3 \times 10^9) = 0.25$ s

clock rate (P1) > clock rate(P2), performance(P1) < performance(P2)

1.12.2 T(P1) = No. instr. × CPI/clock rate

T(P1) = 2.25 3 1021 s

$T(P2)$ 5 $N \times 0.75/(3 \times 10^9)$, then $N = 9 \times 10^8$

1.12.3 MIPS = Clock rate $\times 10^{-6}$/CPI

$MIPS(P1) = 4 \times 10^9 \times 10^{-6}/0.9 = 4.44 \times 10^3$

$MIPS(P2) = 3 \times 10^9 \times 10^{-6}/0.75 = 4.0 \times 10^3$

MIPS(P1) > MIPS(P2), performance(P1) < performance(P2) (from 11a)

1.12.4 MFLOPS = No. FP operations $\times 10^{-6}$/T

MFLOPS(P1) = .4 × 5E9 × 1E-6/1.125 = 1.78E3

1.15

processors	exec. time/ processor	time w/overhead	speedup	actual speedup/ideal speedup
1	100			
2	50	54	100/54 = 1.85	1.85/2 = .93
4	25	29	100/29 = 3.44	3.44/4 = 0.86
8	12.5	16.5	100/16.5 = 6.06	6.06/8 = 0.75
16	6.25	10.25	100/10.25 = 9.76	9.76/16 = 0.61