Solution to CpE 313 HW 01

1.3 This question further explores the effects of Amdahl's Law, but the data given in the question is in a form that cannot be directly applied to the general speedup formula.

a. Because the information given does not allow direct application of Amdahl's Law we start from the definition of speedup:

$$\text{Speedup}_{\text{overall}} = \frac{\text{Time}_{\text{unenhanced}}}{\text{Time}_{\text{enhanced}}}$$

The unenhanced time is the sum of the time that does not benefit from the 10 times faster speedup plus the time that does benefit, but before its reduction by the factor of 10. Thus,

$$\text{Time}_{\text{unenhanced}} = 50\%\ \text{Time}_{\text{enhanced}} + 10 \times 50\%\ \text{Time}_{\text{enhanced}} = 5.5\ \text{Time}_{\text{enhanced}}.$$

Substituting into the equation for Speedup yields

$$\text{Speedup}_{\text{overall}} = \frac{5.5\text{Time}_{\text{enhanced}}}{\text{Time}_{\text{enhanced}}} = 5.5.$$

b. Using Amdahl's Law, the given value of 10 for the enhancement factor, and the value for $\text{Speedup}_{\text{overall}}$ from part (a), we have

$$5.5 = \frac{1}{1 - \text{Fraction}_{\text{enhanced}} + \frac{\text{Fraction}_{\text{enhanced}}}{10}}$$

Solving shows that the enhancement can be applied 91% of the original time.

1.4 a. Speedup $= \dfrac{\text{Number of floating-point instructions DFT}}{\text{Number of floating-point instructions FFT}}$

$$= \frac{n^2}{n \log_2 n}$$

Thus,

n	8	16	32	64	128	256	512	1024
Speedup	2.7	4.0	6.4	10.7	18.2	32.0	56.9	102.4

Also,

$$\lim_{n\to\infty} \text{Speedup} = \lim_{n\to\infty} \frac{n^2}{n \log_2 n} = \infty$$

b. Percent reduction $= 1 - \dfrac{\text{Number of floating-point instructions FFT}}{\text{Number of floating-point instructions DFT}}$

$$= 1 - \frac{1024 \times \log_2 1024}{1024^2}$$

$$= 99\%$$

c. Choosing to include a branch-target buffer in a processor means adding circuitry to the unenhanced design. This increases die size, testing time, and power consumption, making the enhanced processor more costly. Choosing to use an equivalent, asymptotically faster algorithm, such as the FFT, incurs no cost. Thus, the better algorithm will be universally adopted.

1.8 Care in using consistent units and in expressing dies/wafer and good dies/wafer as integer values are important for this exercise.

a. The number of good dies must be an integer and is less than or equal to the number of dies per wafer, which must also be an integer. The result presented here assumes that the integer dies per wafer is modified by wafer and die yield to obtain the integer number of good dies.

Microprocessor	Dies/wafer	Good dies/wafer
Alpha 21264C	231	128
Powe3-II	157	71
Itanium	79	20
MIPS R14000	122	46
UltraSPARC III	118	44

b. The cost per good die is

Microprocessor	$/ good die
Alpha 21264C	$36.72
Powe3-II	$56.34
Itanium	$245.00
MIPS R14000	$80.43
UltraSPARC III	$118.18

c. The cost per good, tested, and packaged part is

Microprocessor	$ / good, tested, packaged die
Alpha 21264C	$64.77
Powe3-II	$78.67
Itanium	$268.33
MIPS R14000	$108.49
UltraSPARC III	$152.18

d. The largest processor die is the Itanium at 300 mm^2. Defect density has a substantial effect on cost, pointing out the value of carefully managing the wafer manufacturing process to maximize the number of defect-free die. The table below restates die cost assuming the baseline defect density from parts (a)–(c) and then for the lower and higher densities for this part.

Itanium	$ / good, tested, packaged die
defect density = 0.5	$268.33
defect density = 0.3	$171.82
defect density = 1.0	$635.83

e. For the Alpha 21264C, tested, packaged die costs for an assumed defect density of 0.8 per cm^2 and variation in parameter α from $\alpha = 4$ to $\alpha = 6$ are $77.53 and $78.59, respectively.