計算機結構 HW1

> author: 111062272 蕭登鴻

> AndeSight version: v3.2

1

(a)

function	CycC	InsC
naive_power_iter()	89	50
naive_power_recur()	218	135
fast_power_iter()	42	27
fast_power_recur()	118	73

Explanation: For the N-th power to any number, naive algorithms have complexity $O(N)$, while fast algorithms have complexity $O(\lg(N))$, and their execution time difference is noticeable even when at small number of N.

(b)

function	Average CPI	Average Execution Time (secs)
fast_power_iter()	1.56	1.4E-8
fast_power_recur()	1.62	3.93E-8

Explanation: Since recursive algorithms have function calls overhead, it is usually slightly slower than their iterative versions, but comes with a much better readability and design simplicity.

(c)

-O0

function	CycC	InsC	CPI
fast_power_iter()	191	87	2.19
fast_power_recur()	267	167	1.60

-O1

function	CycC	InsC	CPI
fast_power_iter()	42	27	1.56
fast_power_recur()	118	73	1.62

Explanation: The cycle count decreases drastically after O1 optimization since the goal of O1 optimization is to reduce the size of compiled code and to boost execution speed without affecting the compile time too much.

(d)

$$\begin{aligned} \text{CPU Time} = \frac{\text{IC}_1 \cdot \text{CPI}_1}{\text{Rate}_1} &= \frac{\text{IC}_2 \cdot \text{CPI}_2}{\text{Rate}_2} \\ \Rightarrow \frac{\text{CycC}_1}{\text{Rate}_1} &= \frac{\text{CycC}_2}{\text{Rate}_2} \end{aligned}$$

fast_power_iter(): 13.64 GHz

fast_power_recur(): 6.79 GHz

2

(a)

Snapdragon 8 Gen 3: 1x 3300 MHz | 3x 3150 MHz | 2x 2960 MHz | 2x 2260 MHz

Apple A17 Pro: 2x 3780 MHz | 4x 2110 MHz

Google Tensor G3 Pro: 1x 2910 MHz | 4x 2370 MHz | 4x 1700 MHz

(b)

	(A) Seconds	(A) Clock Cycles	(B) Seconds	(B) Clock Cycles	(C) Seconds	(C) Clock Cycles
Samsung (Cortex-X4)	0.5013	1654	74.91	247191	61.63	263390
Apple (Everest)	0.6387	2414	71.68	270968	50.57	191150
Google (Cortex-X3)	0.7451	2168	133.3	388000	85.11	247660

(c)

	Samsung Galaxy S24 Ultra	Apple iPhone 15 Pro Max	Google Pixel 8 Pro

	Samsung Galaxy S24 Ultra	Apple iPhone 15 Pro Max	Google Pixel 8 Pro
Samsung Galaxy S24 Ultra	1	0.99987	0.6493
Apple iPhone 15 Pro Max	1.00012	1	0.6494
Google Pixel 8 Pro	1.54	1.5398	1

Where A_{ij} is the performance ratio of $\frac{\text{Perf}_j}{\text{Perf}_i}$.

3

(a)

$$2 = \frac{\text{Perf '}}{\text{Perf}} = \frac{\text{Time}}{\text{Time'}} = \frac{\text{IC} \cdot \overline{\text{CPI}}/\text{RATE}}{\text{IC} \cdot \overline{\text{CPI'}}/\text{RATE}} = \frac{\overline{\text{CPI}}}{\overline{\text{CPI'}}}$$

$$\begin{aligned} \overline{\text{CPI}} &= \frac{1}{S}\left(3 \cdot 100 + 2 \cdot 140 + 5 \cdot 110 + 3 \cdot 55\right), \\ \overline{\text{CPI'}} &= \frac{1}{S}\left(3 \cdot 100 + x \cdot 140 + 5 \cdot 110 + 3 \cdot 55\right) \\ \text{where } S &= 100 + 140 + 110 + 55 = 405 \end{aligned}$$

$$\Rightarrow x = -2.625,$$
$$\text{improvement} = \frac{2-(-2.625)}{2} = 231.25\% \text{ (impossible)}$$

(b)

$$\overline{\text{CPI'}} = \frac{1}{S}\left(3(1-0.28) \cdot 100 + 2(1-0.32) \cdot 140 + 5(1-0.61) \cdot 110 + 3(1-0.64) \cdot 55\right)$$

Then, execution time improvement is calculated by:

$$\frac{T-T'}{T} = 1 - \frac{T'}{T} = 1 - \frac{\text{CPI'}}{\text{CPI}} \approx 0.475 = 47.5\%$$

4

> references: Formula, Communication cost (@eq 6.4)

$$\text{denote } \frac{T}{T'} = S(p,s) = S_{\text{latency}},$$

(a)

$$
\begin{aligned}
S(0.75, 2) &= 1.6 \Rightarrow T' = 8.75\ (ns) \\
S(0.75, 4) &= \frac{16}{7} \approx 2.2857 \Rightarrow T' = 6.125\ (ns) \\
S(0.75, 8) &= \frac{32}{11} \approx 2.9 \Rightarrow T' = 4.8125\ (ns)
\end{aligned}
$$

(b)

$$
\text{denote } \frac{T}{T'} = S'(p, s, \kappa) = \frac{T}{(1-p)T + \frac{p}{s}T + \kappa T} = \frac{1}{(1-p) + \frac{p}{s} + \kappa}
$$

$$
S'(0.75, 8, 0.04) = \frac{800}{307} \approx 2.61
$$

(c)

$$
S'(p, s, \kappa) = S'(p, s, 0.02\lg(s))
$$

$$
\Rightarrow S'(0.75, 8, 0.02\lg(8)) = S'(0.75, 8, 0.06) = \frac{800}{323} \approx 2.47678
$$

5

notations:

$$
\begin{aligned}
D_{pa} &: \text{Defects per area} \\
D_a &: \text{Die area} \\
W_a &: \text{Wafer area} \\
D_{pw} &: \text{Dies per wafer} \\
C_d &: \text{Cost per die} \\
C_w &: \text{Cost per wafer}
\end{aligned}
$$

(a)

$$
0.9 = \frac{1}{\left(1 + \left(\frac{D_{pa} \cdot D_a}{2}\right)\right)^2},\ D_a = \frac{W_a}{D_{pw}},\ W_a = \pi r^2
$$
$$
\text{Where } r = 50/2\ \text{mm},\ D_{pw} = 95
$$

$$
\Rightarrow D_{pa} \approx 5234\ \text{defects}/m^2
$$

(b)

$$
C_d = \frac{C_w}{D_{pw} \cdot \text{Yield}} \approx 0.105263\$
$$

(c)

$$\frac{D'_{pw} - D_{pw}}{D_{pw}} = 0.1,\ D_{pw} = \frac{W_a}{D_a}$$

$$\frac{D_a}{D'_a} = 1.1 \Rightarrow D'_a = D_a/1.1 \approx 1.879 \cdot 10^{-5}\ m^2$$

$$\Rightarrow \text{Yield}' = \frac{1}{\left(1 + \left(\frac{D_{pa} \cdot 1.25 \cdot (D_a/1.1)}{2}\right)\right)^2} \approx 0.887535$$