Karlsruher Institut für Technologie

Institut für Technische Informatik
Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung
Prof. Dr. rer. nat. Wolfgang Karl

Zentralübung Rechnerstrukturen: Low-Power-Entwurf und Leistungsbewertung

Musterlösung zu Aufgabenblatt 2

Low-Power-Entwurf

1. Spannungsabsenkung: 5V $\rightarrow$ 0.8V $\Rightarrow U^2$: 25 $\rightarrow$ 0.64 (Faktor 39.06)

   Frequenzerhöhung: 1MHz $\rightarrow$ 3GHz: Faktor 3000

   Aus $P \sim U^2 * f$ resultiert eine Zunahme der elektrischen Leistung um den Faktor $3000/39.06 \approx 76.8$.

2. Möchte man Prozessoren übertakten, so ist es nötig, dafür zu sorgen, dass die Taktflanken schneller steigen und so schneller gültige Signallevel vorliegen.

   $P_{\text{switching}} = C_{\text{eff}} * U^2 * f$
   Spannungserhöhung führt zu schnellerem Laden von C_{eff} und damit steileren Flanken
   Problematisch: Spannungsbeitrag wird quadratisch verrechnet

3. Aufgrund immer weiterer Verfeinerung der Strukturen spielen mittlerweile die Leckströme eine erhebliche Rolle bei der Leistungsaufnahme. Diese Leckströme werden durch höhere Temperaturen begünstigt.

4. Ermitteln der Schaltwahrscheinlichkeit der gesamten Schaltung $\mathbb{P}_{\text{Schalt Gesamt}}$ über die einzelnen Schaltwahrscheinlichkeiten $\mathbb{P}_{\text{Schalt Gatter}}$ und diese über die Signalwahrscheinlichkeiten $\mathbb{P}_{\text{Gatter}}(Ausgang = 0)$ bzw. $\mathbb{P}_{\text{Gatter}}(Ausgang = 1)$.

   $\mathbb{P}_{\text{Schalt Gatter}} = \mathbb{P}_{\text{g}}(0 \rightarrow 1) + \mathbb{P}_{\text{g}}(1 \rightarrow 0)$
   $\mathbb{P}_{\text{Schalt Gatter}} = \mathbb{P}_{\text{g}'}(0) * \mathbb{P}_{\text{g}}(1) + \mathbb{P}_{\text{g}'}(1) * \mathbb{P}_{\text{g}}(0) = 2 * \mathbb{P}_{\text{g}}(0) * \mathbb{P}_{\text{g}}(1)$
   $\mathbb{P}_{\text{Schalt Gatter}} = 2 * \mathbb{P}_{\text{g}}(1) * (1 - \mathbb{P}_{\text{g}}(1))$

   wegen $\mathbb{P}_{g'}(Ausgang = X) = \mathbb{P}_{g}(Ausgang = X)$ (statistisches Modell!) und $\mathbb{P}(0) = 1 - \mathbb{P}(1)$.

   Betrachtung des ODER-Gatters:

- Signalwahrscheinlichkeit:
  $\mathbb{P}_{Ausgang}(1) = 1 - \mathbb{P}_{Ausgang}(0)$
  $\mathbb{P}_{Ausgang}(1) = 1 - \frac{1}{4} * \frac{3}{4} = \frac{13}{16}$
- Schaltwahrscheinlichkeit:
  $\mathbb{P}_{Schalt} = 2 * \mathbb{P}(1) * (1 - \mathbb{P}(1))$
  $\mathbb{P}_{Schalt} = 2 * \frac{13}{16} * (1 - \frac{13}{16}) = \frac{2*13*3}{16*16} = \frac{39}{128}$

5. Verwenden der Summe der Schaltwahrscheinlichkeiten als Metrik um beide Varianten zu vergleichen.

   Variante 1:

   - Beide linken Gatter: $\mathbb{P}_{\text{UND}}(1) = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$, $\mathbb{P}_{\text{Schalt}} = 2 * \frac{1}{4} * \frac{3}{4} = \frac{3}{8}$.
   - Rechtes Gatter: $\mathbb{P}_{\text{UND}'}(1) = \mathbb{P}_{\text{UND}}(Ausgang = 1) * \mathbb{P}_{\text{UND}}(1) = \frac{1}{4} * \frac{1}{4} = \frac{1}{16}$, $\mathbb{P}_{\text{Schalt}} = 2 * \frac{1}{16} * \frac{15}{16} = \frac{15}{128}$.
   - $Summe_{\text{Schaltw'keiten}} = 2 * \frac{3}{8} + \frac{15}{128} = \frac{111}{128} = 0.8671875$.

   Variante 2:

   - Linkes Gatter: $\mathbb{P}_{\text{UND}}(1) = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$, $\mathbb{P}_{\text{Schalt}} = 2 * \frac{1}{4} * \frac{3}{4} = \frac{3}{8}$.
   - Mittleres Gatter: $\mathbb{P}_{\text{UND}'}(1) = \frac{1}{2} * \frac{1}{4} = \frac{1}{8}$, $\mathbb{P}_{\text{Schalt}} = 2 * \frac{1}{8} * \frac{7}{8} = \frac{7}{32}$.
   - Rechtes Gatter: $\mathbb{P}_{\text{UND}''}(1) = \frac{1}{2} * \frac{1}{8} = \frac{1}{16}$, $\mathbb{P}_{\text{Schalt}} = 2 * \frac{1}{16} * \frac{15}{16} = \frac{15}{128}$.
   - $Summe_{\text{Schaltw'keiten}} = \frac{3}{8} + \frac{7}{32} + \frac{15}{128} = \frac{91}{128} = 0.7109375$.

   Damit kann aus der höheren Summe der Schaltwahrscheinlichkeiten in Variante 1 ein höherer Leistungsverbrauch resultieren, da es wahrscheinlicher ist, dass ein beliebiges Gatter schaltet und somit zusätzliche Leistung aufnimmt. Die zugrundeliegende Schaltstruktur hat jedoch eine geringere Durchlaufzeit (2 Ebenen) im Gegensatz zu Variante 2 (3 Ebenen).

Leistungsbewertung

1. Die Zykluszeit hängt von der Organisation und der Technologie ab.
   Die Anzahl der Instruktionen ist bedingt durch die Befehlssatzarchitektur und die Güte des Compilers.
   Die Zyklen pro Instruktion werden durch die Organisation und die Befehlssatzarchitektur beeinflusst.

2. $f = \frac{i*CPI}{t}, MIPS = \frac{f}{CPI*10^6}$
   $f_A = \frac{3.5*10^6*\frac{7}{5}}{2*10^{-3}} = 2450MHz$
   $MIPS_A = \frac{f_A}{CPI_A*10^6} = \frac{2.45*10^9}{\frac{7}{5}*10^6} = 1750MIPS$

$f_B = \frac{1.5*10^6*\frac{3}{2}}{2*10^{-3}} = 1125MHz$
$MIPS_B = \frac{f_B}{CPI_B*10^6} = \frac{1.125*10^9}{\frac{3}{2}*10^6} = 750MIPS$

Es ist Prozessor B zu wählen, weil

- ohne Berechnung: Gleich schnell in der Abarbeitung (2ms) bei wesentlich weniger ausgeführten Instruktionen (1.5 vs. 3.5 Mio Instruktionen)
- halbe Taktfrequenz ($P \sim U^2 * f$, Fertigung)

3. Benchmark-Berechnung

   - Anzahl Instruktionen
     $i = \sum i_{typ}$
     $= (300 + 75 + 150 + 25) * 10^3 = 550.000$

   - Taktzyklen
     $c = \sum i_{typ} * c_{typ}$
     $= (300 * 1 + 75 * 2 + 150 * 3 + 25 * 4) * 10^3 = 1.000.000$

   - Zykluszeit bei 4GHz Taktfrequenz
     $t = \frac{1}{f} = \frac{1}{4GHz} = 0.25 * 10^{-9}s = 0.25ns$

   - Ausführungszeit
     $t_{exec} = c * t_{cyc}$
     $= 1000 * 10^3 * 0.25 * 10^{-9} = 250 * 10^{-6}s = 250\mu s$

   - CPI
     $CPI = \frac{c}{i} = \frac{1000*10^3}{550*10^3} = \frac{100}{55} = \frac{20}{11} \approx 1.82$

   - MIPS
     $MIPS = \frac{i}{t*10^6} = \frac{550.000}{250} = 2200$

   - MFLOPS: wie MIPS, wobei Anzahl der Befehle und Ausführungszeit nur für Fließkommaberechnung
     $MFLOPS = \frac{75.000}{(75.000*2)*(0.25*10^{-9})*10^6} = \frac{1}{0.5*10^{-3}} = 2000$

4. (vergl. Hennessy and Patterson, Computer Architecture A Quantitative Approach, 4.Auflage, S. 43-44.)

   Es ändern sich nur die Zyklen pro Instruktion, Taktrate und Anzahl der Instruktionen bleiben gleich.

Der unoptimierte CPI-Wert errechnet sich nach:

$$CPI_{base} = \sum_{i=1}^{n} CPI_i * \frac{IC_i}{InstructionCount_{total}} = (4*0,25) + (1,33*0,75) = 2,0.$$

Die Zyklen pro Instruktion mit neuem FPSQR: $CPI_{newFPSQR}$ kann durch Abziehen der gesparten Zyklen erfolgen:

$$\begin{aligned} CPI_{newFPSQR} &= CPI_{base} - 0,02 * (CPI_{oldFPSQR} - CPI_{newFPSQR}) \\ &= 2,0 - 0,02 * (20-2) = 1,64. \end{aligned}$$

Die Alternative mit dem neuen CPI_{FP}-Wert errechnet sich analog zum CPI_{base}:

$$CPI_{newFP} = (0,75*1,33) + (0,25*2,5) = 1,62.$$

Aufgrund des geringeren CPI-Werts bietet sich die Alternative 2 mit den verbesserten Zyklen pro Gleitkommaoperation an.

Berechnung des Gewinns (Speedup) durch die Verwendung der Alternative (b) gegenüber dem vorherigen System (base):

$$\begin{aligned} Speedup_{(b)} &= \frac{CPU\,time_{base}}{CPU\,time_{(b)}} \\ &= \frac{i_{base} * Taktrate_{(b)} * CPI_{base}}{i_{(b)} * Taktrate_{base} * CPI_{(b)}} \\ &= \frac{CPI_{base}}{CPI_{(b)}} \end{aligned}$$

Eingesetzt ergibt sich:

$$Speedup_{(b)} = \frac{2,00}{1,62} \approx 1,23$$

$\rightarrow$ Alternative (b) ist $1,23$-mal schneller als das bisherige System. Die Berechnung des $Speedup_{(a)}$ verbleibt als kleine Übung.

5. a) Der Laufzeitunterschied zwischen dem Base und Peak-Setup ist in den erlaubten Optimierungen zu suchen. Während Base nur konservative Standardoptimierungen erlaubt und gleiche Compileroptionen für alle Benchmarks vorschreibt, erlaubt Peak das aggressive Optimieren für die individuelle Architektur. Für 483.xalancbmk fällt auf, dass die Laufzeitunterschiede vernachlässigbar sind, dies lässt zwei Schlüsse zu:
   - entweder waren die durchgeführten Optimierungen nicht wirkungsvoll,
   - oder weitere Optimierungen wurden nicht angestrebt.

Ein Blick in die Sektion *Peak Optimization Flags* der Webseite verrät, dass außer `basepeak=yes` (welches nur die Anzahl der laufenden Kopien des Programms auf dem System beeinflusst) keine Compileroptimierungen angestrebt wurden. Dies steht im Gegensatz zu allen weiteren Programmen, die für den Peak-Lauf mit aggressiven Compileroptimierungen übersetzt wurden.

b) Die gesuchte Formel lautet: $SPEC_{ratio} = \frac{Referenzzeit_x}{Laufzeit_x\ auf\ Testsystem}$ für einen Benchmark x. Umstellen und einsetzen der $SPEC_{ratio}$ und der Laufzeit für x aus der Tabelle liefert für

$Referenzzeit_{462.libquantum} = 613 * 33,8\,\text{s} = 20719,4\,\text{s}$.

c) Es ist die Ultra Enterprise 2 von Sun Microsystems. Hierauf weist die errechnete Referenzlaufzeit des ausgewählten Benchmarks hin, welche recht gut mit dem Median der Laufzeiten überein stimmen:

Benchmark	$Referenzzeit_{errechnet}$	$Laufzeit_{UltraEnterprise2}$
462.libquantum	20719,4	20704

und auch der $SPEC_{int\_base}2006 = 1.00$ spricht dafür.

6. a) **Bedienzeiten:** $X_i = t_{Zugriff} + t_{Übertragung}$

$X_1 = 12\,\text{ms} + \frac{100\,\text{kB}}{6000\,\text{kB/s}} = 28,67\,\text{ms}$
$X_2 = 10\,\text{ms} + \frac{100\,\text{kB}}{7500\,\text{kB/s}} = 23,33\,\text{ms}$
$X_3 = 8\,\text{ms} + \frac{100\,\text{kB}}{8000\,\text{kB/s}} = 20,5\,\text{ms}$

b) **Maximaler Durchsatz:** $D_{imax} = \frac{1}{X_i}$

$D_{1max} = \frac{1}{28,67\,\text{ms}} = 34,88\,\frac{1}{\text{s}}$
$D_{2max} = \frac{1}{23,33\,\text{ms}} = 42,86\,\frac{1}{\text{s}}$
$D_{3max} = \frac{1}{20,5\,\text{ms}} = 48,78\,\frac{1}{\text{s}}$

Nur Platten mit $D_{max} > A$ können eingesetzt werden, da sonst die Festplatte nicht genügend Zeit hat, um alle Aufträge rechtzeitig zu bedienen. Aufgrund von $A = 40/s$ ist somit nur die Platte 2 einsetzbar.

c) **Auslastung:** $U_i = D/D_{imax} = D * X_i$, wegen Systemsicht gilt hier $D = A$

$U_2 = D * X_2 = 40\,\frac{1}{\text{s}} * 23,33\,\text{ms} = 0,93$, d.h. 93 % Auslastung
$U_3 = D * X_3 = 40\,\frac{1}{\text{s}} * 20,5\,\text{ms} = 0,82$, d.h. 82 % Auslastung

d) **Gesetz von Little:** $Q = W * D$

Q: Anzahl von Aufträgen in der Warteschlange
W: Wartezeit

D: Durchsatz

d.h. $W_i = \frac{Q_i}{D}$, wobei abermals gilt $D = A$ (Systemsicht)
$W_2 = \frac{Q_2}{D} = \frac{3}{40/s} = 75\,ms$
$W_3 = \frac{Q_3}{D} = \frac{2}{40/s} = 50\,ms$

Reaktionszeit des Gesamtsystems aus Warteschlange und Festplatte:

- $Reaktionszeit_i = Wartezeit_i + Bedienzeit_i$
- einsetzen ergibt:

  $Reaktionszeit_2 = 75\,ms + 23,33\,ms = 98,33\,ms$

  $Reaktionszeit_3 = 50\,ms + 20,5\,ms = 70,5\,ms$

Damit ist das System mit Platte 3 vorzuziehen, da es schneller reagiert.