

Betriebssysteme und Rechnerarchitektur
WS 2015/16
LV 3142

Übungsblatt 7
Bearbeitungszeit 2 Wochen
Abgabetermin: 8.02.2016, 4:00 Uhr

Aufgabe 7.1:

Entwickeln Sie einen einfachen Cache-Simulator in C! Das Programm `cachesim.c` soll seine Parameter in folgender Reihenfolge über die Kommandozeile gesteuert verarbeiten:

- Dateiname mit der Zugriffssequenz
- Cachegröße (in Bytes!)
- Assoziativität: 1 = direct mapped, 2 = 2-Wege-Set-Assoziativ, 4 = 4-Wege-Set-Assoziativ, 8 = 8-Wege-Set-Assoziativ
- Nutzungsart: D = Daten-Cache, I = Instruktions-Cache, U = Unified (Beides)

Die Cacheline-Size sei fix auf 16 Bytes beschränkt und (falls erforderlich) sei die Verdrängungsstrategie LRU. Der Cache sei initial leer. Erstellen Sie ein Programm, das basierend auf diesen Parametern die Anzahl der Cache-Hits und die Hit-Rate (Anzahl Hits / Anzahl Zugriffe) bestimmt. Ein Hit kann beim Lesen oder Schreiben gleichermaßen erfolgen.

Im Repository finden Sie die Zugriffssequenzen eines Testbeispiels (`test.trc`) und zweier Programme des SPEC92-Benchmarks (`gcc.trc` und `tomcat.trc`), ausgeführt auf einem MIPS R3000 Prozessor im folgenden Format:

- Das File ist ein ASCII-Textfile. Die Anzahl der Zeilen ist kleiner als 2^{31} . Jede Zeile enthält ein Tupel (Zugriffstyp, Adresse in Hexadezimaldarstellung). Die Zugriffstypen sind:
- 0 read data
- 1 write data
- 2 instruction fetch
- (andere Typen kommen nicht vor bzw. können ignoriert werden)

Die Datei `test.trc` beinhaltet ein vereinfachtes Beispiel mit nur 10 Datenzugriffen. Gegeben ist die Lösung für den 4-Wege-Set-Assoziativen Daten-Cache mit 4 KB Größe (Relevante Zugriffe: 10; Cache-Hits: 3; Hit-Rate: 0.300000):

0 1000	Miss (belegt danach Way #1)
1 100c	Hit
1 2000	Miss (belegt danach Way #2)
1 3000	Miss (belegt danach Way #3)
1 4005	Miss (belegt danach Way #4)
0 4567abcd	Miss (anderer Set, belegt dort Way #1)
0 1000	Hit (aktualisiert LRU)

1 5000	Miss (Verdrängt den ältesten Eintrag 0x200x)
1 4567abcf	Hit (im anderen Set)
1 2000	Miss (Verdrängt 0x300x)

Hinweis zur Entwicklung:

Lesen Sie die Namen der Trace-Dateien über die Kommandozeile ein und geben Sie zunächst die folgenden Kenngrößen aus:

- Die Größe einer Cache-Line
- Anzahl der Cache-Lines
- Anzahl Wege im Cache
- Anzahl Sets im Cache
- zugehöriger Set für jeden relevanten Zugriff
- zugehöriger Tag für jeden relevanten Zugriff
- Anzahl der Gesamtzugriffe (alle Daten- und Instruktionszugriffe)
- Anzahl der für die jeweilige Analyse relevanten Zugriffe (je nach Nutzungsart)

Testen Sie Ihr Programm mit den Trace-Daten und überprüfen Sie die Ergebnisse.

```
studi@poolpc:~/bsra$ ./cachesim test.trc 4096 4 D
cachelinesize: 16
cachelines: 256
ways: 4
sets: 64
...
```

Erweitern Sie danach Ihren Simulator um die Cache-Logik und die Bestimmung weiterer Kenngrößen:

- Anzahl der Cache-Misses
- Anzahl der Cache-Hits
- Hit-Rate

Aufgabe 7.2:

Lassen Sie Ihren Simulator für die folgenden Fälle (Cachegröße, Assoziativität, Nutzungsart) die Hit-Rate bestimmen. Benutzen Sie dazu das Skript `run.sh`. Die Referenzwerte dienen der Kontrolle:

Trace	32K,A1,U	32K,A2,U	32K,A4,U	32K,A8,U	8K,A1,I	8K,A2,I	8K,A1,D	8K,A2,D
tomcatv			0.966370					
gcc								0.960455

Aufgabe 7.3:

Schlagen Sie ein möglichst günstiges Cache-Layout mit bis zu 10 KByte Speicher für den Tomcat-Trace vor. Überlegen Sie zunächst die möglichen Varianten, die insbesondere auch Kombinationen aus dedizierten Daten- bzw. Instruktions-Caches sein können. Simulieren Sie diese Cache-Layouts wählen Sie die optimale Variante für den Tomcat-Trace aus. Dokumentieren Sie Ihre Ergebnisse in der Datei `layout.txt`.

Bewertung:

Aufgabe	Kriterien	Punkte
7.1	Grundfunktionalität:	
	- korrekte Berechnung der abhängigen Größen bei 2-er Potenzen als Cache-Größe	1
	- Einlesen der Datei	1
	- Nutzungsarten U / I / D	2
	- Simulation Unified Cache / Berechnung Set und Tag	3
	- Assoziativität A2, A4, A8	3
	Geforderte Ausgaben	1
7.2	Korrekte Funktionalität	5
7.3	Korrekte, anhand der Messwerte schriftlich begründete Auswahl (Folgefehler nicht bewerten)	4
	Abzüge bei Compiler-Warnungen	(-2)
	Abzüge Lesbarkeit / Kommentare	(-3)
	Gesamt	20