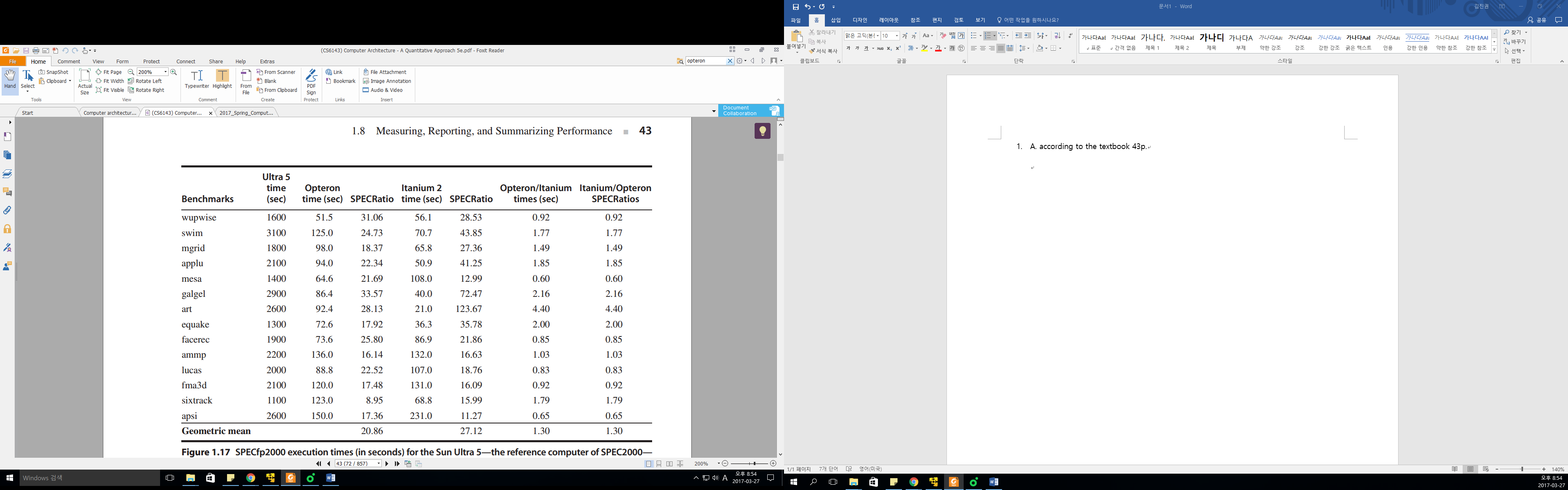
Homework1

20173155

김진권

1. A. according to the textbook 43p. you can see the spec200 benchmark result in itanium2 and opteron. In the chart, you can see the geometric mean of SPECRatio. Itanium2 got a 27.12 and the opteron got a 20.86, so we can say that itanium2 is much faster than opteron.



B. weighted average of execution time ratios for this mix of applications

= (opteron/itanium) 0.6 \* 0.92 (wepwise) + 0.2 \* 1.03 (ammp) + 0.2 \* 0.65 (apsi) = 0.888

1. A.

True Data dependency causes read after write hazard.

1. Data dependency for register R1 from the LD to the DADDI
2. Data dependency for register R1 from the DADDI to the SD
3. Data dependency for register R2 from the DADDI to the DSUB
4. Data dependency for register R4 from the DSUB to the BNEZ

Antidependence (a kind of name dependence) causes write after read hazard

1. Antidependence for register R2 from the SD to the DADDI

Output dependence (a kind of name dependence) causes write after write hazard

1. Output dependence for register R1 from the LD to the DADDI

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 1.LD | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.DADDI |  | IF | stall | stall | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |  |  |
| 3.SD |  |  |  |  | IF | stall | stall | ID | EX | MEM | WB |  |  |  |  |  |  |  |
| 4.DADDI |  |  |  |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |  |
| 5.DSUB |  |  |  |  |  |  |  |  | IF | stall | stall | ID | EX | MEM | WB |  |  |  |
| 6.BNEZ |  |  |  |  |  |  |  |  |  |  |  | IF | stall | stall | ID | EX | MEM | WB |
| 7. |  |  |  |  |  |  |  |  |  |  |  |  |  |  | IF | ID | EX | Flushed |
| 8. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | IF | ID | Flushed |
| 9. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | IF | Flushed |
| 1.LD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | IF |

B. current mips solve the conditional branch in stage 4

Branch가 총 99번 일어나고, 한 번 instruction 전체 cycle이 18 cycle 걸리고, 마지막 cycle에 instruction이 fetch되기 때문에, , 17\*99 +1 = 1684이 된다.

1. A
2. Normal operation – no branch penalty
3. Jumps and calls – branch penalty = 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 jump | Pipe1 | Pipe2 | Pipe3 | Pipe4 |  |  |
| 2 inst(targetX) |  | Pipe1 | flushed | flushed | flushed |  |
| 3 inst(targetO) |  |  | Pipe1 | Pipe2 | Pipe3 | Pipe4 |
|  |  |  |  |  |  |  |

1. Conditional branches when taken – branch penalty = 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 conb(taken) | Pipe1 | Pipe2 | Pipe3 | Pipe4 |  |  |  |
| 2 inst(targetX) |  | Pipe1 |  | flushed | flushed |  |  |
| 3 inst(targetO) |  |  |  | Pipe1 | Pipe2 | Pipe3 | Pipe4 |
|  |  |  |  |  |  |  |  |

1. Conditional branches when not taken – branch penalty = 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 conb(not taken) | Pipe1 | Pipe2 | Pipe3 | Pipe4 |  |  |  |
| 2 inst(targetO) |  | Pipe1 |  | Pipe2 | Pipe3 | Pipe4 |  |
| 3 inst(targetO) |  |  |  | Pipe1 | Pipe2 | Pipe3 | Pipe4 |
|  |  |  |  |  |  |  |  |

Total CPI = 1 + 0.01 \* 1 (jump penalty) + 0.15 \* 0.6 \* 2 (conditional branch when taken) + 0.15 \* 0.4 \* 1 (conditional branch when not taken) = 1.25

B.

1. Normal operation – no branch penalty

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |  |  |
| 1 jump | Pipe1 | Pipe2 | Pipe3 | Pipe4 | Pipe5 | Pipe6 | Pipe7 | Pipe8 | Pipe9 | Pipe10 | Pipe11 | Pipe12 | Pipe13 | Pipe14 | Pipe15 |  |  |  |  |  |
| 2 inst  (targetX) |  | Pipe1 |  |  |  | flushed | flushed | flushed | flushed | flushed | flushed | flushed | flushed | flushed | flushed | flushed | flushed | flushed | flushed |  |
| 3 inst  (targetO) |  |  |  |  |  | Pipe1 | Pipe2 | Pipe3 | Pipe4 | Pipe5 | Pipe6 | Pipe7 | Pipe8 | Pipe9 | Pipe10 | Pipe11 | Pipe12 | Pipe13 | Pipe14 | Pipe15 |

2. Jumps and calls – branch penalty = 4

3.Conditional branches when taken – branch penalty = 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 1 jump | Pipe1 | Pipe2 | Pipe3 | Pipe4 | Pipe5 | Pipe6 | Pipe7 | Pipe8 | Pipe9 | Pipe10 | Pipe11 | Pipe12 | Pipe13 | Pipe14 | Pipe15 |  |  |  |
| 2 inst  (targetX) |  | Pipe1 |  |  |  |  |  |  |  |  | flushed | flushed | flushed | flushed |  |  |  |  |
| 3 inst  (targetO) |  |  |  |  |  |  |  |  |  |  | Pipe1 | Pipe2 | Pipe3 | Pipe4 | Pipe5 | Pipe6 | Pipe7 | Pipe8 |

4.conditional branches when not taken – branch penalty = 8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 1 jump | Pipe1 | Pipe2 | Pipe3 | Pipe4 | Pipe5 | Pipe6 | Pipe7 | Pipe8 | Pipe9 | Pipe10 | Pipe11 | Pipe12 | Pipe13 | Pipe14 | Pipe15 |  |  |  |
| 2 inst  (targetO) |  | Pipe1 |  |  |  |  |  |  |  |  | Pipe2 | Pipe3 | Pipe4 | Pipe5 | Pipe6 | Pipe7 | Pipe8 | Pipe9 |
| 3 inst  (targetX) |  |  |  |  |  |  |  |  |  |  | Pipe1 | Pipe2 | Pipe3 | Pipe4 | Pipe5 | Pipe6 | Pipe7 | Pipe8 |

Total CPI = 1 + 0.01 \* 4 (jump penalty) + 0.15 \* 0.6 \* 9 (conditional branch when taken) + 0.15 \* 0.4 \* 8 (conditional branch when not taken) = 2.33