Homework2

20173155

김진권

1.A.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | unscheduled | | | scheduled | | |
| 1 |  | DADDIU | R4,R1,#800 |  | DADDIU | R4,R1,#800 |
| 2 | foo | L.D | F2,0(R1) | foo | L.D | F2,0(R1) |
| 3 |  | Stall |  |  | L.D | F6,0(R2) |
| 4 |  | Stall |  |  | DADDIU | R1,R1,#8 |
| 5 |  | MUL.D | F4,F2,F0 |  | MUL.D | F4,F2,F0 |
| 6 |  | L.D | F6,0(R2) |  | DADDIU | R2,R2,#8 |
| 7 |  | Stall |  |  | DSLTU | R3,R1,R4 |
| 8 |  | Stall |  |  | Stall |  |
| 9 |  | Stall |  |  | Stall |  |
| 10 |  | Stall |  |  | Stall |  |
| 11 |  | Stall |  |  | Stall |  |
| 12 |  | Stall |  |  | Stall |  |
| 13 |  | ADD.D | F6,F4,F6 |  | ADD.D | F6,F4,F6 |
| 14 |  | Stall |  |  | Stall |  |
| 15 |  | Stall |  |  | Stall |  |
| 16 |  | Stall |  |  | Stall |  |
| 17 |  | Stall |  |  | BNEZ | R3,foo |
| 18 |  | S.D | F6,0(R2) |  | S.D | F6,-8(R2) |
| 19 |  | DADDIU | R1,R1,#8 |  |  |  |
| 20 |  | DADDIU | R2,R2,#8 |  |  |  |
| 21 |  | Stall |  |  |  |  |
| 22 |  | DSLTU | R3,R1,R4 |  |  |  |
| 23 |  | Stall |  |  |  |  |
| 24 |  | Stall |  |  |  |  |
| 25 |  | Stall |  |  |  |  |
| 26 |  | BNEZ | R3,foo (branch가 ID단계에서 해결되기 때문에, 기존 latency + 1이 latency가 된다.) |  |  |  |
| 27 |  | stall | Branch가 resolve될때될 때 기다리는 branch strategy에서는 stall을 1 cycle 추가해야 한다. |  |  |  |
| 26 cycle per element | | | | 17 cycle per element | | |

`

첫번째 instruction DADDIU R4,R1,#800의 경우 loop의 basic block에 포함되지 않기 때문에 고려하지 않았다.

Unscheduled execution time : 26 cycle per element

Scheduled execution time : 17 cycle per element

Scheduled가 unscheduled보다 26/17 = 1.52배 빠르기 때문에, performance를 match해주기 위해서는 clock이 1.52배 더 빨라야 한다. Cpu time = CPI \* IC \* cct이기 때문에 동일한 속도를 내기 위해서는 CPI의 증가만큼 cct를 감소시켜야한다. (IC는 동일 코드이기 때문에 동일 )

B.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Without any stall | | |
| 1 |  | DADDIU | R4,R1,#800 |
| 2 | foo | L.D | F2,0(R1) |
| 3 |  | L.D | F3,8(R1) |
| 4 |  | L.D | F8,16(R1) |
| 5 |  | L.D | F11,24(R1) |
| 6 |  | MUL.D | F4,F2,F0 |
| 7 |  | MUL.D | F5,F3,F0 |
| 8 |  | MUL.D | F9,F8,F0 |
| 9 |  | MUL.D | F12,F11,F0 |
| 10 |  | L.D | F6,0(R2) |
| 11 |  | L.D | F7,8(R2) |
| 12 |  | L.D | F10,16(R2) |
| 13 |  | L.D | F13,24(R2) |
| 14 |  | ADD.D | F6,F4,F6 |
| 15 |  | ADD.D | F7,F5,F7 |
| 16 |  | ADD.D | F10,F9,F10 |
| 17 |  | ADD.D | F13,F12,F13 |
| 18 |  | DADDIU | R1,R1,#32 |
| 19 |  | S.D | F6,0(R2) |
| 20 |  | S.D | F7,8(R2) |
| 21 |  | DSLTU | R3,R1,R4 |
| 22 |  | S.D | F10,16(R2) |
| 23 |  | S.D | F13,24(R2) |
| 24 |  | BNEZ | R3,foo |
| 25 |  | DADDIU | R2,R2,#32 |
| 24 cycle per loop (4 element per loop) | | | |

첫번째 instruction DADDIU R4,R1,#800의 경우 loop의 basic block에 포함되지 않기 때문에 고려하지 않았다.

1. times loop must be unrolled. Execution time : 24/4 = 6 per element

2.A

1.assume when issuing the instruction into reservation station, first set the busy flag in that reservation station. (at the same time, it occupy the reservation station)

2.assume reservation station is free at write CDB station, and an instruction can get in there at the same cycle.

3. L.D and S.D enter the integer reservation station (they need to calculate the address) and L.D&S.D buffer at issue. And they free the reservation station at write CDB (also an instruction can get in there at the same cycle).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| iteration | instruction | issues | executes | Memory access | Write CDB | comment |
| 1 | L.D F2,0(R1) | 1 |  | 2 | 3 | [1-2] int reservation station  [1-2] L.D buffer |
| 1 | MUL.D F4,F2,F0 | 2 | 4 |  | 19 | [3-3] waiting F2  [2-18] FP mult reservation station |
| 1 | L.D F6,0(R2) | 3 |  | 4 | 5 | [3-4] int reservation station  [3-4] L.D buffer |
| 1 | ADD.D F6,F4,F6 | 4 | 20 |  | 30 | [5-5] waiting F6  [5-19] waiting F4  [4-29] FP add reservation station |
| 1 | S.D F6,0(R2) | 5 |  | 31 |  | [6-30] waiting F6  [5-31] int reservation station  [5-31] S.D buffer |
| 1 | DADDIU R1,R1,#8 | 6 | 7 |  | 8 | [6-7] int reservation station |
| 1 | DADDIU R2,R2,#8 | 7 | 8 |  | 9 | [7-8] int reservation station |
| 1 | DSLTU R3,R1,R4 | 8 | 9 |  | 10 | [8-9] int reservation station |
| 1 | BNEZ R3,foo | 9 | 11 |  |  | [10-10] Waiting R3  [9-11] int reservation station |
| 2 | L.D F2,0(R1) | 10 |  | 12 | 13 | [11-11] Waiting branch  [10-12] int reservation station  [10-12] L.D buffer |
| 2 | MUL.D F4,F2,F0 | 11 | 19 |  | 34 | [12-18] FP multiplier busy  [12-13] waiting F2  [11-33] FP multi reservation station |
| 2 | L.D F6,0(R2) | 12 |  | 13 | 14 | [12-13] int reservation station  [12-13] L.D buffer |
| 2 | ADD.D F6,F4,F6 | 13 | 35 |  | 45 | [14-14] waiting F6  [14-34] waiting F4  [13-44] FP add reservation station |
| 2 | S.D F6,0(R2) | 14 |  | 46 |  | [15-45] Waiting F6  [14-46] int reservation station  [14-46] S.D buffer |
| 2 | DADDIU R1,R1,#8 | 15 | 16 |  | 17 | [15-16] int reservation station |
| 2 | DADDIU R2,R2,#8 | 16 | 17 |  | 18 | [16-17] int reservation station |
| 2 | DSLTU R3,R1,R4 | 17 | 18 |  | 19 | [17-18] int reservation station |
| 2 | BNEZ R3,foo | 18 | 20 |  |  | [19-19] Waiting R3  [18-20] int reservation station |
| 3 | L.D F2,0(R1) | 19 |  | 21 | 22 | [20-20] waiting branch  [19-21] int reservation station  [19-21] L.D buffer |
| 3 | MUL.D F4,F2,F0 | 20 | 34 |  | 49 | [21-22] waiting F2  [21-33] FP multiplier busy  [20-48] FP multi reservation station |
| 3 | L.D F6,0(R2) | 21 |  | 22 | 23 | [21-22] int reservation station  [21-22] L.D buffer |
| 3 | ADD.D F6,F4,F6 | 22 | 50 |  | 60 | [23-23] waiting F6  [23-49] waiting F4  [22-59] FP add reservation station |
| 3 | S.D F6,0(R2) | 23 |  | 61 |  | [24-60] waiting F6  [23-61] int reservation station  [23-61] S.D buffer |
| 3 | DADDIU R1,R1,#8 | 24 | 25 |  | 26 | [24-25] int reservation station |
| 3 | DADDIU R2,R2,#8 | 26 | 27 |  | 28 | [25-25] int reservation station full  [26-27] int reservation station |
| 3 | DSLTU R3,R1,R4 | 28 | 29 |  | 28 | [27-27] int reservation station full  [28-29] int reservation station |
| 3 | BNEZ R3,foo | 30 | 31 |  |  | [29-29] int reservation station full  [30-31] int reservation station |

[reservation station status]





B

1.assume when issuing the instruction into reservation station, first set the busy flag in that reservation station. (at the same time, it occupy the reservation station)

2.assume reservation station is free at write CDB station, and an instruction can get in there at the same cycle.

3. L.D and S.D enter the integer reservation station (they need to calculate the address) and L.D&S.D buffer at issue.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| iteration | instruction | issues | executes | Memory access | Write CDB | comment |
| 1 | L.D F2,0(R1) | 1 |  | 2 | 3 | [1-2] int reservation station  [1-2] L.D buffer |
| 1 | MUL.D F4,F2,F0 | 1 | 4 |  | 19 | [2-3] waiting F2  [1-18] FP mult reservation station |
| 1 | L.D F6,0(R2) | 2 |  | 3 | 4 | [2-3] int reservation station  [2-3] L.D buffer |
| 1 | ADD.D F6,F4,F6 | 2 | 20 |  | 30 | [3-4] waiting F6  [3-19] Waiting F4  [2-29] FP add reservation station |
| 1 | S.D F6,0(R2) | 3 |  | 31 |  | [4-30] Waiting F6  [3-31] int reservation station  [3-31] S.D buffer |
| 1 | DADDIU R1,R1,#8 | 3 | 4 |  | 5 | [3-4] int reservation station |
| 1 | DADDIU R2,R2,#8 | 4 | 5 |  | 6 | [4-5] int reservation station |
| 1 | DSLTU R3,R1,R4 | 4 | 6 |  | 7 | [4-6] int reservation station  [5-5] Int busy |
| 1 | BNEZ R3,foo | 5 | 8 |  |  | [6-7] waiting R3  [5-8] int reservation station |
| 2 | L.D F2,0(R1) | 6 |  | 9 | 10 | [7-8] Waiting bnez  [6-9] int reservation station  [6-9] L.D buffer |
| 2 | MUL.D F4,F2,F0 | 6 | 11 |  | 26 | [7-10] Waiting F2  [6-25] FP mult reservation station |
| 2 | L.D F6,0(R2) | 7 |  | 10 | 11 | [9-9] int busy  [7-10] int reservation station  [7-10] L.D buffer |
| 2 | ADD.D F6,F4,F6 | 7 | 27 |  | 37 | [8-26] Waiting F4  [8-11] Waiting F6  [7-36] FP add reservation station |
| 2 | S.D F6,0(R2) | 9 |  | 38 |  | [8-8] int reservation station full  [10-38] waiting F6  [9-38] int reservation station  [9-38] S.D buffer |
| 2 | DADDIU R1,R1,#8 | 10 | 11 |  | 12 | [9-9] int reservation station full  [10-11] int reservation station |
| 2 | DADDIU R2,R2,#8 | 11 | 12 |  | 13 | [10-10] int reservation station full  [11-12] int reservation station |
| 2 | DSLTU R3,R1,R4 | 12 | 13 |  | 14 | [11-11] int reservation station full  [12-13] int reservation station |
| 2 | BNEZ R3,foo | 13 | 15 |  |  | [12-12] int reservation station full  [14-14] waiting R3  [13-15] int reservation station |
| 3 | L.D F2,0(R1) | 14 |  | 16 | 17 | [15-15] waiting bnez  [14-16] int reservation station |
| 3 | MUL.D F4,F2,F0 | 19 | 20 |  | 35 | [14-18] FP mult reservation station full  [19-34] FP mult reservation station |
| 3 | L.D F6,0(R2) | 19 |  | 20 | 21 | [19-20] int reservation station  [19-20] L.D buffer |
| 3 | ADD.D F6,F4,F6 | 20 | 36 |  | 46 | [21-21] waiting F6  [21-35] waiting F4  [20-45] FP add reservation station |
| 3 | S.D F6,0(R2) | 20 |  | 47 |  | [21-46] Waiting F6  [20-47] int reservation station  [20-47] S.D buffer |
| 3 | DADDIU R1,R1,#8 | 21 | 22 |  | 23 | [21-22] int reservation station |
| 3 | DADDIU R2,R2,#8 | 23 | 24 |  | 25 | [21-22] int reservation station full  [23-24] int reservation station |
| 3 | DSLTU R3,R1,R4 | 25 | 26 |  | 27 | [23-24] int reservation station full  [25-26] int reservation station |
| 3 | BNEZ R3,foo | 27 | 28 |  |  | [25-26] int reservation station full  [27-28] int reservation station |

[reservation station status]



3.

Correlating predictor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PC | Last outcome | Table entry | prediction | result | Correct? | update |
| 454 | T | 454%4=2+T  => 4 | T | T | yes | T |
| 543 | T | 543%4=3+T  =>6 | NT with one misprediction | NT | yes | NT |
| 777 | NT | 777%4=1+NT  =>3 | T | NT | no | T with one misprediction |
| 543 | NT | 543%4=3+NT  =>7 | NT | NT | yes | NT |
| 777 | NT | 777%4=1+NT  =>3 | T with one misprediction | NT | no | NT |
| 454 | NT | 454%4=2+NT  =>5 | T | T | yes | T |
| 777 | T | 777%4=1+T  =>2 | NT | NT | yes | NT |
| 454 | NT | 454%4=2+NT  =>5 | T | T | yes | T |
| 543 | T | 543%4=3+T  =>6 | NT | T | no | NT with one misprediction |

Final misprediction rate = 3/9 => 33%

Local predictor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PC | Last outcome | Table entry | prediction | result | Correct? | update |
| 454 | T,T | 454%2=0+T,T  => 0 | T with one misprediction | T | yes | T |
| 543 | T,T | 543%2=1+T,T  =>4 | T | NT | no | T with one misprediction |
| 777 | T,NT | 777%2=1+T,NT  =>5 | T with one misprediction | NT | no | NT |
| 543 | NT,NT | 543%2=1+NT,NT  =>7 | NT | NT | yes | NT |
| 777 | NT,NT | 777%2=1+NT,NT  =>7 | NT | NT | yes | NT |
| 454 | T,T | 454%2=0+T,T  =>0 | T | T | yes | T |
| 777 | NT,NT | 777%2=1+NT,NT  =>7 | NT | NT | yes | NT |
| 454 | T,T | 454%2=0+T,T  =>0 | T | T | yes | T |
| 543 | NT,NT | 543%2=1+NT,NT  =>7 | NT | T | no | NT with one misprediction |

Final misprediction rate = 3/9 => 33%

4.

A. buffer에 발견되면, 그 다음 instruction을 바로 얻을 수 있기 때문에, 만약 ID부터의 pipeline이 두 개의 instruction을 parallel하게 처리 가능하면, penalty가 발생하는 것이 아니라 이득이 발생하게 된다. 따라서, penalty는 -1로 pipeline의 성능향상을 일으킨다.

B. 우리가 관심있는 unconditional branch의 경우만 고려한다면,

1) 변형된 btb : penalty : 0.05\*(0.9\*-1 (hit penalty = -1 (pipeline에 이득)) + 0.1\*2(buffer miss penalty)) = - 0.035 만큼 이득이 있다.

2) btb가 없는 것과 비교할 경우, performance gain을 얻기 위해서는 hit rate = x라 하면,

0.05\*(-x + (1-x)\*2) < 0 => 2 < 3\*x => x > 0.666 이상이면 performance gain을 얻는다.