Last Modified: 11 May 2014

IS2202 - Computer Systems Architecture

CPU Architecture

Bonus assignment 4

-Submitted by

Tamilselvan Shanmugam tamsha@kth.se

1. ISAs

1.a Why JVM is stack based machine?

JVM is designed as platform independent. If JVM is register based, it has to know about the available register that makes it platform or architecture dependent.

Generated code size is less as the operands are extracted from the bottom of the stack[1].

1.b source of the 2nd operand in accumulator based machine

The source of second operand is given in the instruction.

```
ADD #Mem_Location
```

The above instruction says ADD the value available in the memory location to the accumulator value and store it in the accumulator.

2. Hazards

2.a Structural Hazard

Structural hazard occurs when more than one pipelined instruction are trying to access the same hardware component.

Stalling pipeline is a way to avoid control hazard.

2.b Control Hazard to Data Hazard

Transforming control hazard to data hazard or avoiding branch instructions to increase the basic block size enables compiler to work more on optimization. Predicated execution is a technique to do this.

Special registers called predicate registers are employed for this operation. It holds 1-bit value either TRUE or FALSE. Sequence of instructions can be executed if the predicate register is true or executed instructions can be committed only if predicate register is true[2].

```
if(A == B){
     X = 10;
     Y = Y * Y;
}
```

After predicated execution:

```
(P2 = (A == B))
if P2 then X = 10;
if P2 then y =Y*Y;
```

3.a RAW Hazard:

```
1) ADD R1, R2, R3;  #R1 = R2 + R3

2) MUL R4, R1, R5;  #R4 = R1 X R5
```

In a simple 5-stage pipeline, first instruction is passed through the stages Fetch, Decode, and Execute and waiting to go via Memory and Write back stages.

By this time, 2nd instruction might have done Fetch operation. But couldn't complete decode stage, since the previous instruction has not written the result in R1. So pipeline has to stall until the value is written to R1.

Hardware technique:

Forwarding is a technique to bring the results to ALU as soon as it's computed rather than waiting till Write back stage. ALUs output is fed as one of the ALU operands. MUX can choose which value to operate.

3.b Why can’t WAR and WAW:

WAR	WAW
1) ADD R3, R1, R2; 2) ADD R1, R5, R6;	1) ADD R1, R2, R3; 2) ADD R1, R6, R7;

WAR

In a 5-stage pipeline, R1 is read in second stage for the first instruction. The very next instruction will write the R1 value at the final stage. So, first instruction always read the correct value of R1.

WAW

Both the instruction does the write at the 5th stage. At least they have one cycle gap between those writes and there is no dependency.

So WAR, WAW hazards will not happen in 5-stage pipeline.

3. Instruction scheduling

4. VLIW vs Superscalar

VLIW:

Every instruction has more than one operation to do. All the operands are specified in the instruction and they are sent to right functional unit. If there are no parallel instructions in accordance with the available hardware, compiler inserts NOP.

Superscalar:

More than one instruction are taken, decoded and dispatched to the proper execution units. Reorder buffer does the job of maintaining program order to ensure the correctness.

5. a Reservation station:

Reservation stations are meant to overcome WAR and WAW data hazards by renaming registers and looks weather all operands are available for execution. Instructions are sent to Reservation stations and the station knows the operands. It waits till the operand become available in case of any data dependency and allows instruction to be executed[3].

5.b Reorder Buffer

Reorder buffer has an extra stage called "commit". It guarantees that the commits will happen in the program order i.e the same order as the instructions are dispatched.

This avoids erroneous execution of the program[4].

6. Precise Exception

Precise exceptions are implemented with the help of Reorder buffer in processors that implements Tomasulo's algorithm.

Reorder buffer retire instructions in the program order. Even an instruction is completed out of order, the commits are never been made until all the previous instruction are retired. When exception occurs, reorder buffer commits only the instructions that are previous to the exception and maintains precise exception.

4. Branch prediction

7. 1-bit Branch History Table (BHT)

Program Counters lower bits are used to index a branch history table. BHT stores the action taken last time. When program encounters branch instruction, it looks for the BHT about the last branch decision. It predicts present branch decision and executes the next instruction. When the prediction goes wrong, BHT table entry is updated.

8. 2-bit Branch prediction

As name suggests, it uses 2-bits to store the last branch decision and changes its state accordingly.

2-bit prediction changes decision only if two consecutive predictions were wrong. It remembers the previous decision and previous to previous decision. In general 2-bit branch prediction gives more accurate results than 1-bit branch prediction.

8. Branch Target Buffer (BTB)

Branch Target Buffer stores most significant bit of PC, target address of the branch instruction and previous branch decision.

When a program attempts to run an un-conditional branch, if it is available in BTB, control knows it's gonna be taken and directly goes to target address and start processing target instruction.

So, the un-conditional branch is executed in 0 cycles if it is available in BTB and showcases greatest performance improvement. This is called branch folding.

References:

[1] “virtual machine - Why is the JVM stack-based and the Dalvik VM register-based? - Stack Overflow.” [Online]. Available: http://stackoverflow.com/questions/2719469/why-is-the-jvm-stack-based-and-the-dalvik-vm-register-based. [Accessed: 01-May-2014].

[2]“Predicated Instructions.” [Online]. Available: http://www.csd.ijs.si/courses/trends/tsld034.htm. [Accessed: 10-May-2014].

[3]“Reservation stations - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/Reservation_stations. [Accessed: 02-May-2014].

[4]“Re-order buffer - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/Re-order_buffer. [Accessed: 02-May-2014].