Lab Report 9

Computer Architecture Lab

201651015 Dipansh Khandelwal

201651008 Aman Singh

8th April 2019

Questions and Answers

1. **What is the role of simulators in processor design?**
    - Lowering cost by evaluating hardware designs without building physical hardware systems.
    - Enabling access to unobtainable hardware.
    - Increasing the precision and volume of computer performance data.
    - Introducing abilities that are not normally possible on real hardware such as running code backwards when an error is detected or running in faster-than-real-time.

2. **Why is it advantages to have several different simulators?**
    - One of the primary advantages of simulators is that they are able to provide users with practical feedback when designing real world systems.
    - This allows the designer to determine the correctness and efficiency of a design before the system is actually constructed.
    - Consequently, the user may explore the merits of alternative designs without actually physically building the systems.
    - By investigating the effects of specific design decisions during the design phase rather than the construction phase, the overall cost of building the system diminishes significantly.

3. **For the four branch prediction schemes ' taken|perfect|bimod|comb ', describe the predictor. Your description should include:**
    - **What information the predictor stores (if any)?**
        - In the architecture the predictor stores all the last predictions made and the corresponding output after making those predictions. Using this data it further decides how and what prediction to make in any situation it faces from then.
    - **How the prediction is made?**
        - The prediction is made using the information stored about all the last predictions made and the corresponding output after making those predictions. It goes with the specific choice which have the best throughput in that situation referring to the information stored before.

4. **What is out-of-order execution?**
    - Out-of-order execution, is a technique used in most high-performance microprocessors to make use of cycles that would otherwise be wasted by a certain type of costly delay.
    - The key concept of OoO processing is to allow the processor to avoid a class of delays (termed: "stalls") that occur when the data needed to perform an operation are unavailable.

5. **What is the difference between scoreboarding and Tomasulo?**
    - Control & buffers distributed with Function Units vs. centralized in scoreboard; called “reservation stations”.
        - instructions schedule themselves
    - Registers in instructions replaced by pointers to reservation station buffer.
        - scoreboard => registers primary operand storage
        - Tomasulo => reservation stations as operand storage
    - HW renaming of registers to avoid WAR, WAW hazards.
        - Scoreboard => both source registers read together (thus one could not be overwritten while we wait for the other).
        - Tomasulo => each register read as soon as available.
    - Common Data Bus broadcasts results to all FUs RS’s (FU’s), registers, etc. responsible for collecting own data off CDB.
    - Load and Store Queues treated as FUs as well.

Program Behaviour(Instruction Profiling)

Benchmark	Load	Store	Uncond branch	Cond branch	Integer computation	Fp computation
Anagram	23.75	9.66	5.82	18.45	42.30	0.00
Go	19.67	6.30	3.18	12.74	58.11	0.00
Compress	5.58	59.20	1.14	7.47	25.02	1.59
Applu	20.88	5.16	0.27	3.95	48.26	21.47
Mgrid	0.18	14.17	0.02	14.37	71.26	0.00
Swim	24.09	6.92	2.57	3.46	38.54	24.42
Pearl	26.30	17.88	5.80	12.68	37.09	0.25
gcc	25.80	13.38	4.81	15.51	40.49	0.01

Branch Predictors (**Branch Address Prediction Rate i.e., address-hits/updates**)

Benchmark	Not taken	Taken	Bimod
Anagram	0.5920	0.6480	0.9536
Go	0.5184	0.6811	0.8233
Compress	0.2492	0.8831	0.9730
Applu	0.4005	0.6627	0.8034
Mgrid	0.0182	0.9830	0.9919
Swim	0.5292	0.8977	0.9818
Pearl	0.6476	0.6661	0.9252
gcc	0.5994	0.6372	0.8697