# CSCI 540 Assignment 3

### Name:Rohansai Alahari

Cwid:50307287

1. What do MIR, MPC, MAR, and MDR do in Mic-1 microarchitecture? (10%)

**MIR**- (Micro-Instruction Register) holds the current microinstruction, whose bits drive the control signals that operate the data path

**MPC**- (Micro-Program Counter) is the address register of the control store.

**MAR-** (Memory Address Register) is a 32-bit register linked to the main memory It contains memory references in the form of word addresses. MAR only has one input signal (input from C) and multiple memory operations (read and write).

**MDR-** is Memory Data Register, a 32-bit register connected to the main memory. It contains data words to/from memory references.

Ans:

the MIR- (Micro-Instruction Register), whose bits regulate the control signals that run the data route.The current microinstruction is stored in mir

The control store's address register is called MPC- (Micro-Program Counter).

the MAR- (Memory Address Register). It has word addresses that serve as memory references. Only one input signal (input from C) and numerous memory operations are available for MAR (read and write).A 32-bit register connected to the main in  memory is called MAR

MDR-—Memory Data Register—is attached to the main memory. It includes data words to/from memory references.

1. What do A Bus, B Bus, and C Bus do in Mic microarchitecture?

Ans:

1. An internal processor design with three buses: two operand buses and one result bus

2. to do away with the requirement to fill the H register

(a) Include the A-Bus as a new data path bus.

(b) Enlarge the MIR such that an A-Bus decoder and multiplexer can be added.

(c) The shifter and C bus are driven by the ALU's output.

(d) Registers can be written using C bus values.

3. Six control lines are used to manage the ALU.

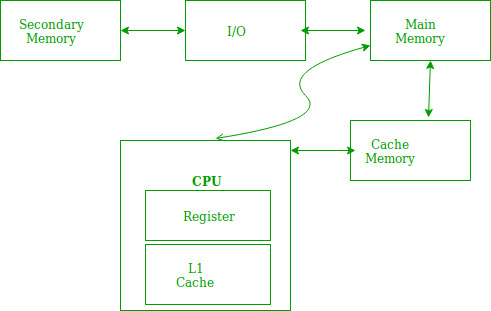
Bus A and Bus B inputs are enabled by ENA and ENB, respectively.

input from bus A is inverted by INVA.

3Where is the cache physically located? What kinds of localities do cache rely on? Briefly explain the localities.

Ans:

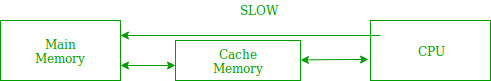
The term "locality of reference" describes a phenomena when a computer program frequently accesses the same group of memory locations over an extended period of time. The tendency of a computer program to retrieve instructions with nearby locations is referred to as locality of reference. In a program, loops and subroutine calls serve as the primary examples of the locality of reference property.

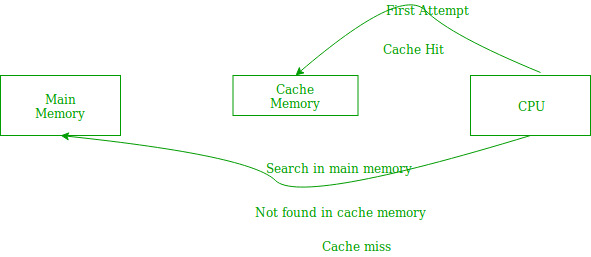


1The set of instructions that make up a loop are constantly referred to when there are loops in a program.

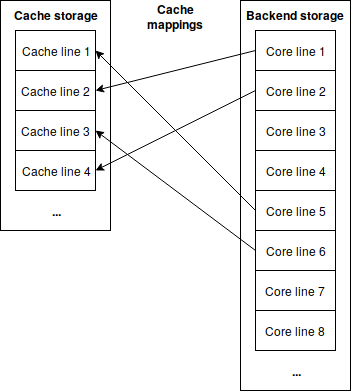
2When a subroutine is called, the set of instructions is periodically fetched from memory.

3Data items are also localized, which results in repeated references to the same data item.

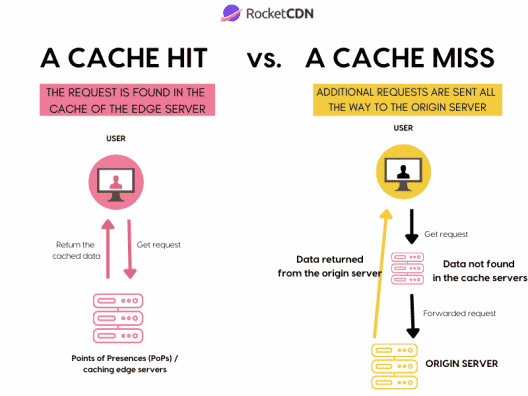


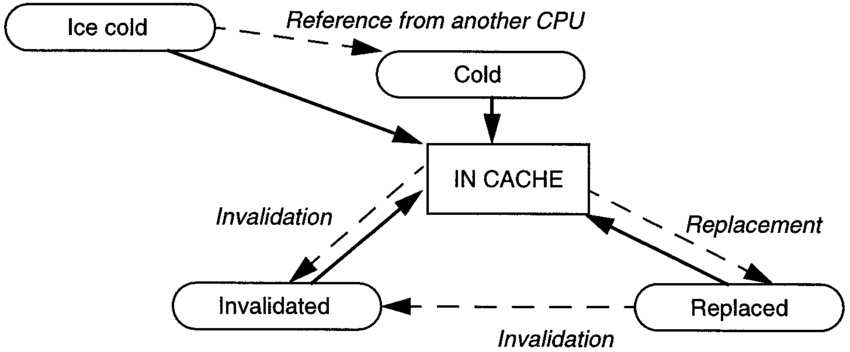
 4What is a cache line? What is a cache hit? What is a cache miss? What will happen when there is a cache miss?.

Cache line:The smallest amount of data that can be mapped into a cache is a cache line. Every mapped cache line has a core line, which corresponds to a corresponding area on a backend store, as its partner.



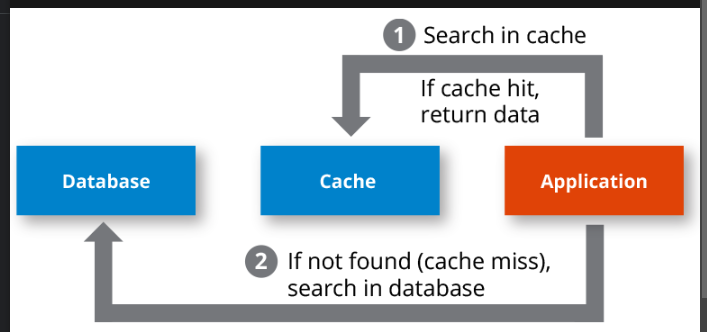
Cache hit:When material from your website is successfully provided from the cache, this is known as a cache hit. When the data is located and read, it is regarded as a cache hit since the tags are quickly sought in the memory. When content is successfully served from the cache rather than the server, this is known as a cache hit.

 cache miss:-

A cache miss occurs when a system or application requests data from a cache but the requested data is not currently stored in the cache memory. In contrast, a cache hit results in the successful retrieval of the requested data from the cache.

What will happen when cache miss happens:-

When a cache miss happens, the system or application continues to search for the information in the underlying data store, lengthening the request. The system may typically write the data to the cache, increasing latency once more; however, this latency is typically offset by cache hits on other data.



1. If on the same machine, the instructions can have different lengths, what are the advantage(s)? What are the potential problem?

Ans:

If the lengths of the instructions on the same machine differ, then:

Advantage:-

-If the instructions are being performed sequentially, then if the processor initially receives the longer instruction, then because the shorter instructions will be available, the processor will be able to pause (if necessary) the longer instructions and execute the shorter ones first.

By doing this, the processor wouldn't be overworked, and users would enjoy a productive experience.

Problem:-

- Due to the existence of instructions of various lengths, it is possible that the processor will receive the longer instructions in series while scheduling the shorter instructions at the end of the queue.

This could hinder the processor's ability to perform efficiently.

A methodical approach of figuring out what might go wrong with a plan under development is called a potential problem analysis (PPA). The severity of the consequences and likelihood of occurrence are used to rank the problem sources. Both preventative measures and backup plans are created.

6.Convert the following infix notations to postfix notations.

1. A×B+C÷D
2. A+B+C+D
3. A+B×(C-D)

Ans:-

AXB+C%D

ABX+C%D

ABX+CD%

ABXCD%+

A+B+C+D

AB++C+D

AB+C++D

AB+C+D+

A+BX(C-D)

A+BX(CD-)

A+B(CD-)X

AB(CD-)X+

7.On a machine, an instruction is always 16 bits long, and there are 16 registers.

1. Briefly explain the idea of expanding opcode. (5%)
2. Design a scheme to support 7 three-address instructions, 31 two-address instructions, 14 one-address instructions, and 32 zero-address instructions.

Ans:

(1) The expanding opcode strategy is used to vary the opcode size in order to accommodate different address forms, such as the 1 address instruction format and the 2 address instruction format. The instruction size will remain constant.

(2) There will be 216 encodings for this instruction because it is 16 bits long. Here, the size of the address field is important and is not asked about. Because of this, 7\*(2addressfieldsize) encodings for 7three-address instructions are possible.

For each of the 31 two-address instructions, there will be a need for 31\*(2addressfieldsize)\*(2addressfieldsize) encodings.

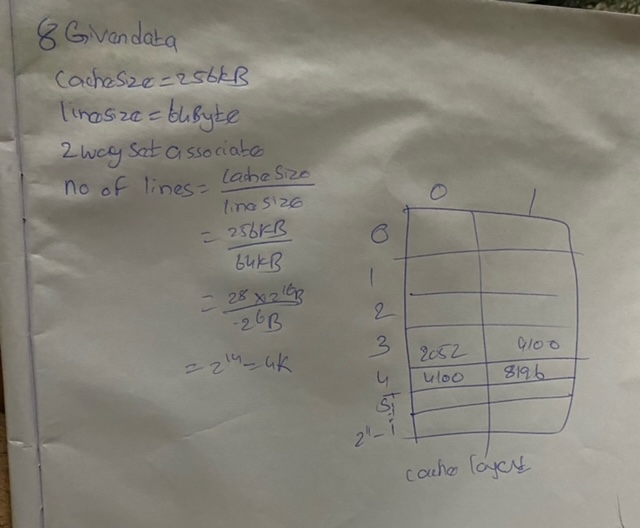
8.Assume that the cache size is 256kB, and each cache line is 64 Bytes. (10%)

1. Let us assume this is a **Two**-way associative cache. How many **cache sets** are there?
2. Let us assume the following memory blocks need to be accessed:

Memory Blocks #4100, 8196, 2052, 8196, 2052, 4100

If the cache is initially empty, what is the cache hit/miss rate?

Ans:



Total no. of session Cache is p'2memory blocks : 4100, 8196, 2052, 8196, 2052, 400  
mapping function = N mods =  
J  
memory  
block  
Creon.  
rest  
© 400 mod 2048 = 4 Miss  
@  
2196 mod 2o48 = 4miss  
@  
2052 mod 2048 = 4mIsS  
2057  
4460  
woe  
8146  
now set o is full, to replace one of memory block,wewilluse  
page replacement algorithm  
Me; LRU Least ReLeNt  
wed)  
(4) 9146 mod 2048 = 4HIt  
Cache  
layout  
G 2052 mod 2048 = 4 Hit  
©  
4100 mad 2048 = 4 muss  
Total  
Hits = 2  
Total  
mIsse1=4  
HIT Ratio = 7 = = 3733\*  
mICs Ratio  
= 66.66 V

9Given this piece of code fragment: (10%)

for (int x = 0; x < 10; x++)

{

if ( x % 4 >= 2)

cout << "OK" << endl;

}

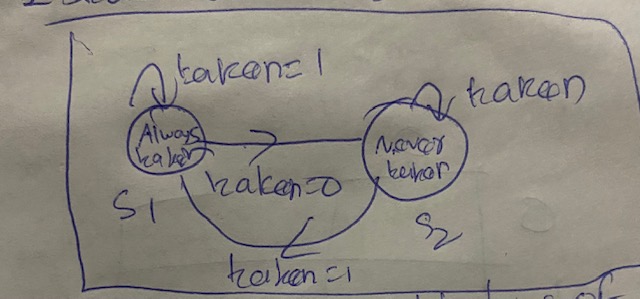
Assuming the branch prediction by default is “TAKEN”,

1. What is the accuracy of branch prediction of the **cout statement** when we use a 1-bit branch history?

What is the accuracy of branch prediction of the **coutstatement** when we use a 2-bit branch history?

Ans:

If we use 1 bit branch history then we will have 2 states and they are as below:



Thus, if (x% 4 >= 2) and x has values between 0 and 9, we have.

for x=0 , 0 % 4 =0 And since it is not greater than 2, the branch won't be taken as the default forecast is made.

that it will fail because we are at the taken stage.

The forecast is now "branch not taken."

For x=1, 1% 4 =1 and it is not greater than 2, so the branch won't take as we stand at this point.

not taken stage, thus it will move on.

The prediction won't change at this time, and the branch won't be taken.

Branch will be taken if x=2 because 2% 4 = 2 and it is greater than 2, however we are now at

not taken stage, hence it will not succeed.

The forecast will now shift to the branch chosen.

branch will be taken because for x=3, 3% 4 = 3 and it is greater than 2, and we are now at

taken stage, so it will move forward.

The prediction won't alter going forward and will be at the chosen branch.

Branch won't take if x=4 since 4% 4 = 0 and it is not greater than 2, and we are already at

therefore it will fail at this stage.

The forecast will now read "branch not taken."

For x=5, 5% 4 = 1 and it is not greater than 2, so the branch will not be taken, and we are now at

not taken stage, thus it will move on.

The prediction won't alter any longer, and the branch will not be taken.with x=6, 6% 4 = 2 and it is greater than 2, causing the branch to take, and we are currently atnot taken stage, hence it will not succeed.

The forecast will now shift to the branch chosen.Given that x=7, 7% 4 = 3, and it is greater than 2, the branch will take, and since we are currently in the taken stage, it will pass.

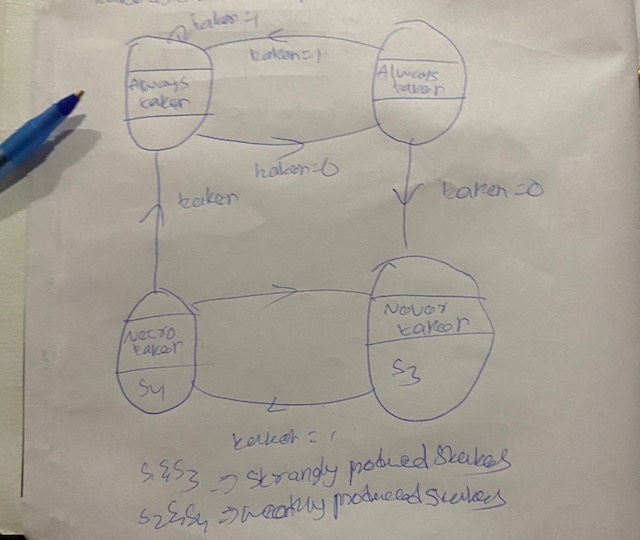
The prediction won't alter going forward and will be at the chosen branch.

Because 8% 4 = 0 for x=8 and it is not greater than 2, the branch won't take and we are now in the taking step, therefore it will fail.

The prediction will now change and the branch will not be taken.

x=9,9% 4=1, and it is not greater than 2.

1. If we use 2 bit branch history then we will have 4 states and they are as below:



taken Always use two tokens. tatin = 1 tablet taken over 3 seconds, strongly advised States As a result, we have if (x% 4 >= 2) and the values of x range from 0 to 9 for x=0. Since 0% 4 = 0 and it is not >= 2, the branch won't take since the default prediction is taken, which means we are at stage S1 and the attempt will fail.

Now that we are at taken stage S2, the prediction has changed to branch taken S2. However, since x=1 and 1% 4 =1 and it is not >= 2, the branch won't take and it will fail.

The forecast will now be for S3 to not be taken.

for x=2 , 2 % 4 =2 and it is >= 2 so branch will take but we are presently at

not taken S3 stage hence it will fail.

Now the prediction will change to branch taken S4.

for x=3 , 3 % 4 = 3 and it is >= 2 so branch will take and we are presently at

taken S4 stage hence it will fail.

Now the prediction will change and it will be at branch taken S1.

for x=4 , 4 % 4 = 0 and it is not >= 2 so branch wont take and we are presently at

taken S1 stage hence it will fail.

Now the prediction will change to branch taken S2.

for x=5 , 5 % 4 = 1 and it is not >= 2 so branch wont take and we are presently at

taken S2 stage hence it will fail.

Now the prediction will change and it will be at branch not taken S3.

for x=6 , 6 % 4 = 2 and it is >= 2 so branch will take and we are presently at

not taken S3 stage hence it will fail.

Now the prediction will change to branch not taken S4.

for x=7 , 7 % 4 = 3 and it is >= 2 so branch will take and we are presently at

not taken S4 stage hence it will fail.

Now the prediction will change and it will be at branch taken S1.

for x=8 , 8 % 4 = 0 and it is not >= 2 so branch wont take and we are presently at

taken S1 stage hence it will fail.

Now the prediction will change and it will be branch taken S1.

for x=9 , 9 % 4 = 1 and it is not >= 2 so branch wont take and we are presently at

taken S2 stage hence it will fail.

Now the prediction wont change and it will be at branch not taken S3.

10Assume we have 8 registers, R0~R7, and we have a pipeline of 6 stages:

Instruction Fetch (IF), Decode & Issue (DI), Operands Fetch (OF), Execution (EX), Write Back (WB), and Commitment (CO). Each stage needs exactly 1 cycle to finish its work.

Also assume that the pipeline supports forwarding, which means the result of WB can be forwarded to OF.

Given the following piece of instructions:

R1 = R0 + R2

R5 = R1 + R4

R1 = R5 – R6

R3 = R0 + R7

1. Identify all the data dependencies and their types. (5%)
2. How many cycles do we need if we run the instructions in a 6-stage pipeline that does not support forwarding, register renaming, and out-of-order execution? (DI stage only decodes the instruction but does not rename its registers.)(5%)

How many cycles do we need if we run the instructions in the 6-stage pipeline that supports forwarding, register renaming, and out-of-order execution?

Ans:

To decrease the dependent on nearby instructions and boost pipelined execution performance, register renaming and out-of-order execution might be used.

The instructions are as follows:

oR1 = R0 + R2, oR5 = R1 + R4, oR1 = R5 - R6, and oR3 = R0 + R7.

Since there is a write after read dependency between the first and fourth instructions, the resister R0 in the first instruction can be changed to R1, and the third and fourth instructions can then be rearranged. The amended instruction set is displayed below.

oR1 = R0 + R2, oR5 = R1 + R4, oR1 = R5 - R6, and oR3 = R0 + R7.

The pipeline can be shown as follows:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| R1 = R1 + R2 | IF | II | OF | EX | WB | CO |  |  |  |
| R5= R1+ R4 |  | IF | II | OF | EX | WB | CO |  |  |
| R1= R5 – R6 |  |  | IF | II | OF | EX | WB | CO |  |
| R3= R0 + R7 |  |  |  | IF | II | OF | EX | WB | CO |

**Number of cycles required in the pipelined execution is 9.**