Assignment 2:
Computer Architecture

Yazeed AlKhalaf

Course: CIS 304 - Computer Architecture
Instructor: Dr. Adeel Baig

Date: 21 Apr, 2024

Contents

1 Question 1:

A cache consists of 128 lines. The main memory contains 8192 blocks of 256 words each. Design the address format if the cache is

(a) Direct Mapped

(b) Associative Mapped

(c) Set Associative with four-line per sets

1.1 Shared Information:

- $cacheLines = 128$ cache line
- $blocks = 8192$ block
- $words = 256$ word
- One memory address bit length: $\log_2(blocks * words) = \log_2(8192 * 256) = \log_2(2097152) = 21$ bits

1.2 Part A: Direct Mapped

For the direct mapped cache, we need to know three things:

- Required cache lines bit length: $\log_2(cacheLines) = \log_2(128) = 7$ bits
- Required word offset bit length: $\log_2(words) = \log_2(256) = 8$ bits
- Required tag bit length is the remaining bits length: $21 - 7 - 8 = 6$ bits

The design is: **6 Tag Bits — 7 Cache Line Bits — 8 Word Offset Bits**

1.3 Part B: Associative Mapped

Associative mapped cache means the cache can store any block in any line. Therefore, we only need to know the word offset bit length:

- Required word offset bit length: $\log_2(256) = 8$ bits
- Required tag bit length is the remaining bits length: $21 - 8 = 13$ bits

The design is: **13 Tag Bits — 8 Word Offset Bits**

1.4 Part C: Set Associative with four-line per sets

Set associative means that the cache lines are split into sets, and each set has some lines, in our case 4 lines, and those are associative which means 4 blocks can exist at the same set.

And here since our lines are split into sets of four, we need to get the number of sets to be able to calculate the "cache set" bit length. To calculate the number of sets, we do the following: $(cacheLines \div 4)$. The 4 here is inferred from the "four-line per set" part.

Taking all of that in mind, we can get the following values:

- Number of cache sets: $128 \div 4 = 32$ cache sets
- Required "cache sets" bit length: $\log_2(32) = 5$ bits
- Required word offset bit length: $\log_2(256) = 8$ bits
- Required tag bit length is the remaining bits length: $21 - 5 - 8 = 8$ bits

The design is: **8 Tag Bits — 5 Cache Set Bits — 8 Word Offset Bits**

2 Question 2:

2.1 Introduction

THe goal of learning 6502 assembly language is purely academic, since we won't use it in our jobs, or perhaps our businesses. Since assembly is the lowest level language before the raw machine code, so it gives good understanding of the underlying things we we work with a fancy high-level language.

6502 assembly language was made with humans reading and writing it in mind. So learning it is easier and more fun than learning X86, as Nick Morgan claims. Also it is easier in general since the intrsuction set for 6502 is made up of 56 instructions. On the other hand, the x86 number of instructions can reach up to a 1000 instructions, and it is made for compilers to compile from high-level languages to it.

With that in mind, we can start the journey of exploring 6502 assembly language!

2.2 Our first program

2.3 Registers and flags

2.4 Instructions