### COSC 2440 – Computer Organization and Architecture – Spring 2020 - Kevin B Long

# 

# Final Exam

Monday, May 4, 2020

Up to 5 submissions accepted, last one graded. 100 points total.

Verify that your name and ID are correct.

Name: Xena Toumajian PeopleSoft ID: 1662518

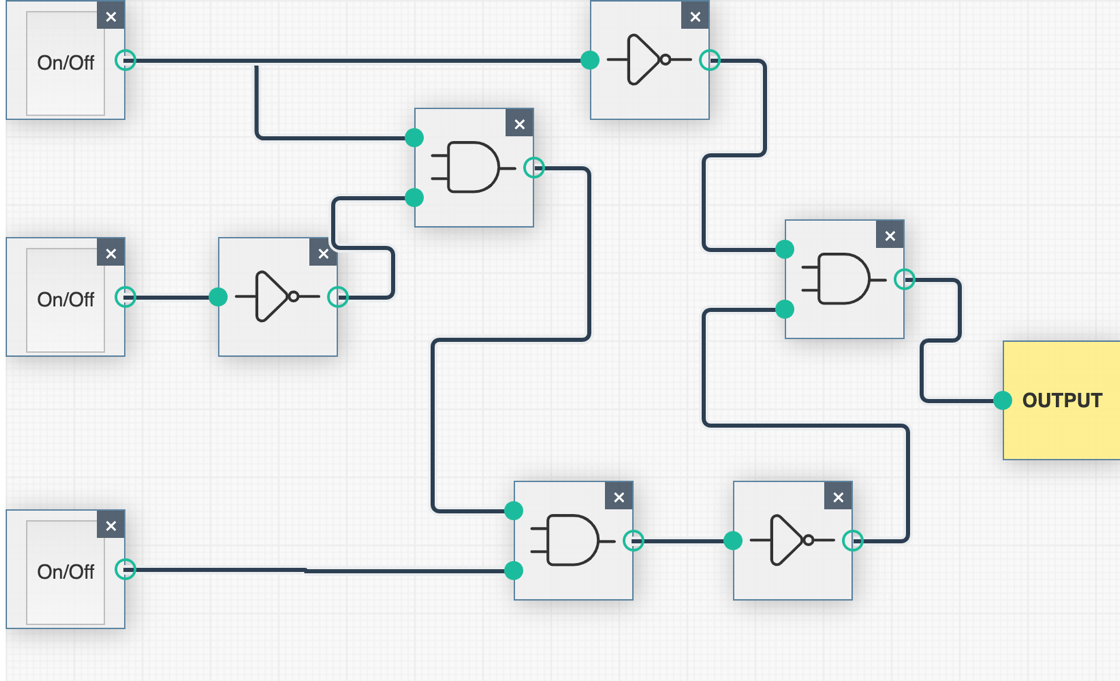
1. Truth tables.
2. Draw a truth table for the following equation. Use only “0” and “1” for the final column entries.

¬(a\*¬b\*c)\*¬a

You are permitted to use <https://web.stanford.edu/class/cs103/tools/truth-table-tool/> to learn how to generate your results, and work out how to express the equation above.

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c | **(¬(a ∧ (¬b ∧ c)) ∧ ¬a)** |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

1. Using the tool at <https://academo.org/demos/logic-gate-simulator/> , and draw a circuit for your equation. Paste the resulting image here:



1. Consider the following rules for simplifying equations:

|  |  |  |
| --- | --- | --- |
| AND rules | OR rules | Name of rule |
| AB = BA | A+B=B+A | Commutative |
| AA=A | A+A=A | Idempotent |
| A • ¬A = 0 | A + ¬A = 1 | Complement |
| A • 1 = A | A + 0 = A | Identity |
| A • 0 = 0 | A + 1 = 1 | Annulment |
| ¬(¬A) = A |  | Double Negation |
| A(BC)=(AB)C | A+(B+C)=(A+B)+C | Associative |
| A(B+C)=AB+AC | A+BC=(A+B)(A+C) | Distributive |
| A+(AB)=A | A(A+B)=A | Absorptive 1 |
| A + (¬A • B) = A + B |  | Absorptive 2 |
| ¬(A + B) = ¬A • ¬B |  | DeMorgan’s 1 |
| ¬(A • B) = ¬A + ¬B |  | DeMorgan’s 2 |

Use those rules to simplify the following equation.

Label each step with the rule used. There is a drop-down menu, but it is not functional, just type your answer after the end of each step.

1. AB(A+B) + BC original equation
2. AAB + ABB + BC Distributive
3. AB + AB + BC Idempotent
4. AB + BC Idempotent
5. B(A + C) Distributive
6. Cache Simulation

Visit the block replacement simulator at this site: <http://www.ecs.umass.edu/ece/koren/architecture/Cache/frame1.htm>

Start with the following settings:

* Set the cache size to 16
* Set the # of Sets to 4
* Leave the Replacement Policy to LRU
* Make sure you disable the Limit Query.
* If they are not already displayed for you below, copy the series in the Cache Sequence column from the input sheet and paste it here:

1 60 13 1 17 41 11 28 42 12 41 35 54 27 20 36 43 12 35 57 27 5 25 12 8 1 3 18 20 53 53 28 40 40 35 2 5 10 9 40 40 6 1 45 15 26 48 39 10 60 13 15 22 21 20 35 44 15 47 48 28 33 53 14 42 38 34 56 35 6 33 46 37 17 47 48 34 27 19 33 14 45 29 44 28 3 57 29 54 37 60 44 36 30 10 31 44 45

After the first cycle…

1. how many times did the cache have the requested location in memory available in cache?

31

1. How many times did it not?

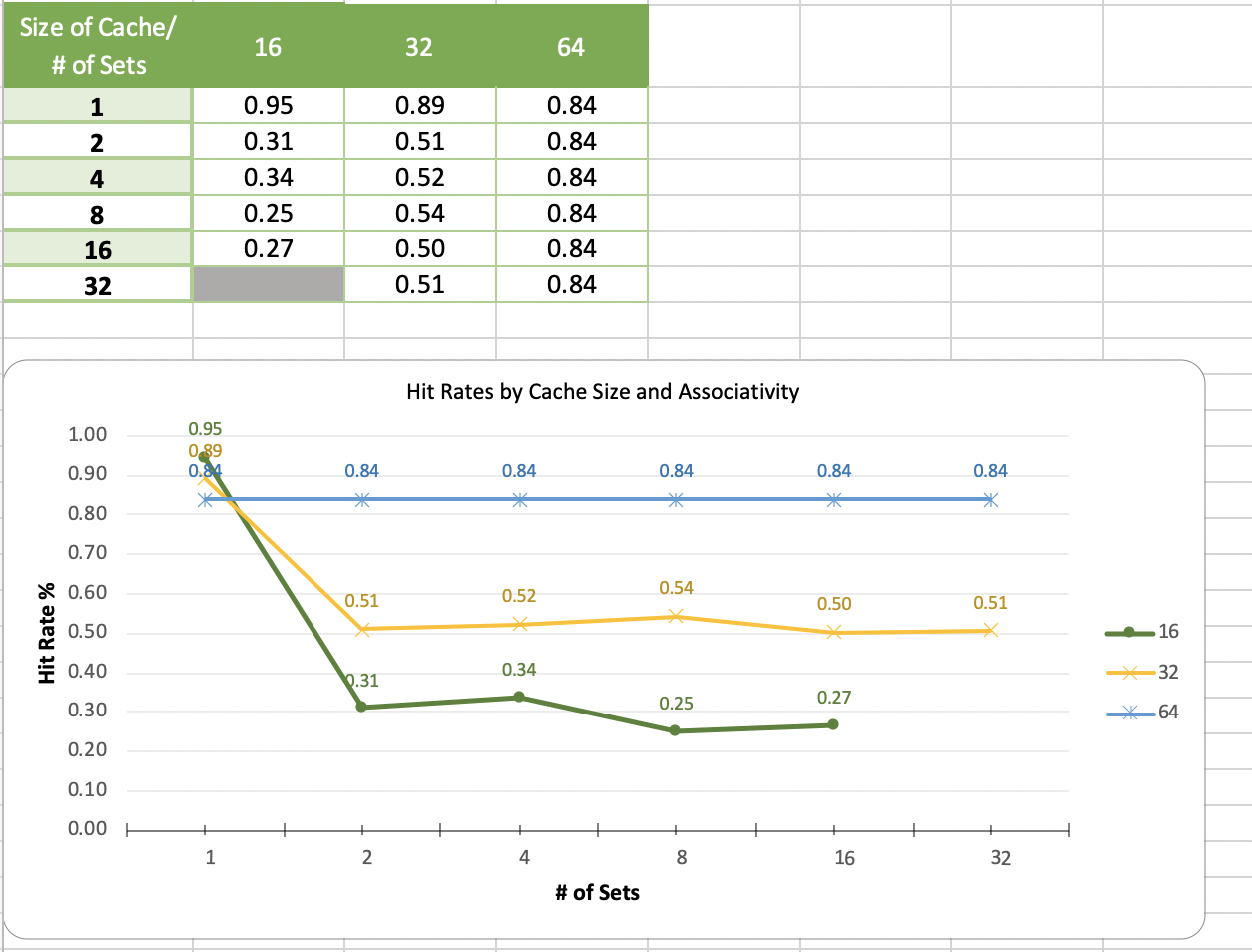
67

1. If you change to a fully-associative cache, what is the hit rate? 83.67%

* Ensure that Repeat is set to 3 cycles

Next, download the spreadsheet attached to the homework – it is on Blackboard and on the class drive, named “Cache Simulator for Final.xlsx”. There is also a version in Google Sheets [here](https://drive.google.com/file/d/1pTvJ4P-KxJ-P-PZgsQgLgQpN1F47Bk0s/view?usp=sharing). These will open in read-only mode, so you can copy it to your own location to edit.

Note: this is a bit different than the homework – here we will change both the # of sets and the size of the cache. Only fill in answers from the 3rd (last) cycle.

1. Run the required simulations and enter your answers into the spreadsheet, let the graph generate, and screen grab it and paste it below.
2. All other things the same, which cache would you expect would have the most hits? A larger one? Or a smaller one?

Larger  Smaller

Is that what your data shows?

Yes  No

Bonus: If no, what is the explanation? You will need to look carefully at the Cache Query Results tables in the simulator and try and reproduce their rate calculations.

1. AMAT - Average Memory Access Time

Consider the cache system shown below, measured like the homework – cumulatively.

Complete the following table to calculate AMAT for a system with 5 levels of memory: 3 cache plus RAM and Disk.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mem Hierarchy | Isolated Time (ns) | Cumulative Time (ns) | Global Hit Rate | Weighted Time (ns) |
| L1 Cache | 5 | 5 | 0.888 | 4.44 |
| L2 Cache | 11 | 16 | .048 | 0.768 |
| L3 Cache | 30 | 46 | 0.029 | 1.334 |
| RAM | 82 | 128 | 0.028 | 3.584 |
| Disk | 9650 | 9778 | 0.00700000000000001 | 68.446 |
|  |  |  | AMAT | 78.572 |

1. Hamming Code

Calculate the 12-bit Hamming Code to be transmitted given the following byte as data input:

01010100

You can use the table below, but the most important bits are p1, p2, p4 and p8.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Preparing for Transmission | | | | | | | | | | | | | | | |
|  |  | Data to Send | 01010100 | | | |  |  |  |  |  |  |  |  |
|  |  | bit position | p1 | p2 | d1 | p4 | d2 | d3 | d4 | p8 | d5 | d6 | d7 | d8 |
|  |  |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| Coverage Matrix | p1 | 0 | 0 |  | 0 |  | 1 |  | 1 |  | 0 |  | 0 |  |
| p2 | 0 |  | 0 | 0 |  |  | 0 | 1 |  |  | 1 | 0 |  |
| p4 | 0 |  |  |  | 0 | 1 | 0 | 1 |  |  |  |  | 0 |
| p8 | 1 |  |  |  |  |  |  |  | 1 | 0 | 1 | 0 | 0 |
| p16 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| p32 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| p64 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Data with Codes | | | **0** | **0** | **0** | **0** | **1** | **0** | **1** | **1** | **0** | **1** | **0** | **0** |