STUDENT NUMBER	A								

NATIONAL UNIVERSITY OF SINGAPORE

FINAL QUIZ

(Semester I: 20xx/20xx)

CG2028 – COMPUTER ORGANIZATION

November 20xx – Time Allowed: 1 Hour

INSTRUCTIONS TO CANDIDATES

1. This paper contains **TWO (2)** questions and comprises **THREE (3)** printed pages, including this page.
2. Write your student number in the box above, as well as on ALL pages with answers.
3. Each question carries 25 marks and has multiple parts. All questions and parts thereof are compulsory. Answer **ALL** questions.
4. Do NOT interleave answers to parts from different questions.
5. This is an **OPEN BOOK** examination. Only hard-copy references are allowed. Electronic references are not allowed.
6. Programmable calculators are **ALLOWED.**
7. Unless otherwise specified, all assembly instructions in Question 1 refer to **ARMv7E-M** instructions. The microarchitecture in Question 2 refers to the microarchitecture detailed in Lectures 4-6.
8. Return this booklet to the invigilator at the end of the exam. Do not tear off pages from this booklet.

Q.1 (a) The formula for the determining the volume of a sphere with radius *r* is

$$V = \frac{4}{3}\pi r^3$$

Write the main body of an ARMv7-M assembly language function that calculates the sum of the volumes of *N* spheres with radii r_i, where $i = 1, .., N$.

The assembly language function does not need to deal with floating point numbers and only needs to handle integer values. Assume that all floating-point values have been scaled by a scaling factor α to get scaled integer approximations before the assembly language program is executed.

The scaled integer approximations of the r_i values are stored in consecutive 4-byte words in main memory starting from location RADIUS. The value of *N* is stored at location N and the result should be stored in a 4-byte word at location RESULT.

(10 marks)

(b) Explain how a main program and the assembly language function in part Q.1 (a) can be used to calculate the sum of the volumes of spheres with the radii values shown in Table Q.1(b).

Table Q.1(b)

i	r_i
1	9.962
2	4.671
3	3.293
4	0.939
5	1.745

(4 marks)

(c) What is the largest unsigned integer value that a word of length 4 bytes can hold?

(3 marks)

(d) Assuming that $r_i \leq 10$ and $N \leq 5$, determine an appropriate scaling factor α such that overflow does not occur and the loss of precision in the numerical values is minimized. The same scaling factor α should be used for all floating-point quantities in the expression shown in part Q.1 (a) above.

For the radii values shown in Table Q.1(b), complete Table Q.1(d) using the scaling factor α that you have determined.

Table Q.1(d)

Word address	Content
RADIUS	
RADIUS+4	
RADIUS+8	
RADIUS+12	
RADIUS+16	

(8 marks)

Q.2 (a) Many real-world data memory modules require a MemEnable signal to be asserted when either a data memory *read* operation *or* a data memory *write* operation is to be performed. This MemEnable signal is generated by the control unit of the processor, just like other control signals. Derive an expression for generating this MemEnable signal.

(5 marks)

(b) A cache is 2-way set associative and follows the LRU replacement strategy. The cache has a total of 4 blocks, with each block composed of 16 bytes. The main memory is byte-addressable and can hold 4096 bytes.
The following locations are accessed by a processor after coming out of reset - 0xA08, 0xA18, 0xA24, 0xA00, 0xA54, 0xA40, 0xA38, 0xA0C, 0xA58.
Draw a diagram illustrating where each incoming block will be placed in the cache. Indicate clearly if a particular access results in a hit or a miss, and the block that is replaced in event of a miss.

(8 marks)

(c) The 5-stage pipelined microarchitecture in Chapter 6 is modified such that instead of controlling the multiplexer at the PC input using PCSrcW, PCSrcE is used. All other connections remain unchanged.

(i) Explain clearly why the design will not work even if you are allowed to insert NOPs as necessary.

(ii) How will you modify the microarchitecture such that it can execute correctly provided an appropriate number of NOPs are inserted while retaining the PCSrcE connection as described above?

(iii) How many NOPs will need to be inserted after a Branch instruction for correct operation in the corrected design?

(iv) What are the potential advantage(s) and drawback(s) of the corrected design?

Brief text-based explanations are sufficient for (i)-(iv) above, a diagram is not expected.

(4+3+2+3 = 12 marks)

END OF PAPER