**CS2100 Computer Organisation**

**AY2021/22 Semester I**

**Assignment 1 [ANSWER SHEET]**

|  |  |
| --- | --- |
| **FULL NAME:** | **Toh Jia-Yih Ronald** |
| **STUDENT ID:**  E.g., <AxxxxxxY> | **A0199937B** |
| **TUTORIAL GROUP:** | **2** |

**QUESTION 0.** **SUBMISSION INSTRUCTIONS (3 MARKS)**

|  |  |
| --- | --- |
| 1. Ensure that you name your file <AxxxxxxxY>.pdf, where AxxxxxxxY is your matric number. (1 mark) | **Y / N** |
| 1. Ensure that you submit your assignment as a single PDF file. (1 mark) | **Y / N** |
| 1. Ensure that your assignment submission has your tutorial group number, student ID and name | **Y / N** |

**QUESTION 1. COMPLEMENT NUMBER SYSTEMS (10 MARKS)**

|  |  |
| --- | --- |
| Q1.a | -m = 4n - m |
| Q1.b | (i) 100011111012 (ii) 1100111002 |
| Q1.c | (i) 1013314 (ii) 121304 |
| Q1.d | (i) -1149 = 46 – 1149 = 294710 = 12320034  (ii) -412 = 45 – 412 = 61210 = 1212104 |
| Q1.e | 1149 + (-412) = 1013314 + 1212104 = 0232014 = 73710 (Verified) |

**QUESTION 2. REAL NUMBERS (11 MARKS)**

|  |  |
| --- | --- |
| Q2.a | (i) 2m-1 - 1 |
| (ii) -2m-1 |
| (iii) 2 - (16-m) |
| Q2.b | |  |  |  |  | | --- | --- | --- | --- | | **m** | **Most positive integer** | **Most negative integer** | **Smallest positive value** | | 4 | **7** | **-8** | **2-12** | |
| Q2.c | |  |  |  | | --- | --- | --- | | **Most positive value** | **Most negative value** | **Smallest positive value** | | **1.996 x 264** | **-1.996 x 264** | **1.000 x 2-63** | |
| Q2.d | Advantage: Floating-point number representations allow a much larger range of values to be expressed as compared to fixed-point representation.  Disadvantage: Floating-point number loses precision as it will be rounded off in large numbers. |

**QUESTION 3. C and Assembly Programming (8 MARKS)**

|  |  |
| --- | --- |
| Q3.a | xor $s0, $s0, $t1 |
| Q3.b | 2 + 31\*5 + 2 = 159 |
| Q3.c | include <math.h>  int t0 = data;  int t1 = (int) pow(2, 31);  while (t0 != 0){  int t2 = t0 & 1;  if (t2 == 1){  s0 = s0 ^ t1;  }  t0 = t0 >> 1;  }  s0 = s0 ^ t1; |

**QUESTION 4. INSTRUCTION ENCODING (8 MARKS)**

|  |  |
| --- | --- |
| Q4.a | Add 1 to every element in array. 10 elements, base address of array at $3. |
| Q4.b | 62 |
| Q4.c | (Provide encodings only for the four instructions in **bold and underline.**)   |  |  |  | | --- | --- | --- | | **Label** | **Instruction** | **Hexadecimal Encoding** | |  | **addi $4, $3, 40** | **0x30640028** | |  | addi $5, $3, 0 |  | | loop: | lw $6, 0($5) |  | |  | addi $6, $6, 1 |  | |  | **sw $6, 0($5)** | **0xACA60000** | |  | addi $5, $5, 4 |  | |  | **slt $6, $5, $4** | **0x00A4302A** | |  | **bne $6, $zero, loop** | **0x14C0FFF9** | |