NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

EXAMINATION FOR

Semester 1 AY2011/12

CS2100 – COMPUTER ORGANISATION

Nov 2011 **Time allowed: 2 hours**

Your Matriculation Number:

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of **SIX (6)** questions and comprises **TWENTY (20)** printed pages including this page.
2. This is an **OPEN BOOK** examination. You may use any approved calculators but not any PDA or laptop, especially those capable of external connectivity or communication.
3. Answer *all* questions. Note that the full mark for each question is different.
4. Write your answers on *this* **QUESTION AND ANSWER SCRIPT**. Answer only in the space given. Any writing outside this space will not be considered. No other submission is allowed.
5. Fill in your Matriculation Number with a pen, clearly on every page of this QUESTION AND ANSWER SCRIPT.
6. You may use pencil to write your answers.
7. At the end of the examination, please check to ensure that your script has all the pages properly stapled together.
8. Note that when a number is written as "0xNNNN" it means that "NNNN" is in base 16.

Total Score | | /100

QUESTION 1 (15 marks)

(1a) For the following 3-bit standard Gray code, fill in the *even* parity bit. (2 marks)

ANSWER:

A	B	C	Even Parity (P)
0	0	0	
0	0	1	
0	1	1	
0	1	0	
1	1	0	
1	1	1	
1	0	1	
1	0	0	

(1b) You are to design a 4-bit synchronous counter that counts the above 4-bit sequence looping back to 000 after reaching 100, using two T-flip-flops (for b_2 and b_0) and two JK flip-flops (for b_1 and P). First, fill in the control signals in the table below.

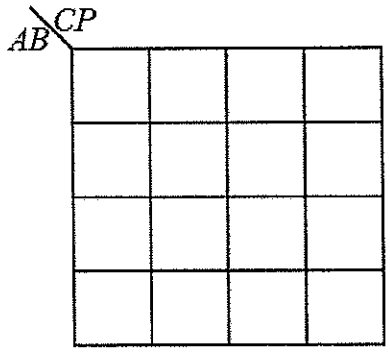
(5 marks)

ANSWER:

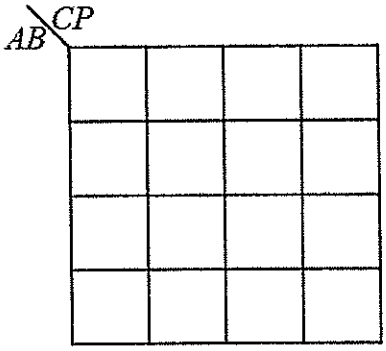
Current State				Next State				T_A	J_B	K_B	T_C	J_P	K_P
A	B	C	P	A	B	C	P						
0	0	0		0	0	1							
0	0	1		0	1	1							
0	1	1		0	1	0							
0	1	0		1	1	0							
1	1	0		1	1	1							
1	1	1		1	0	1							
1	0	1		1	0	0							
1	0	0		0	0	0							

(1c) Draw a K-map for each of the control signal and give the minimum product-of-sum expression for each. (5 marks)

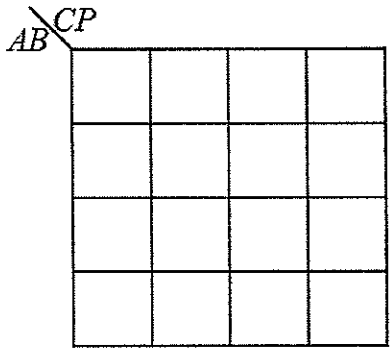
ANSWER:



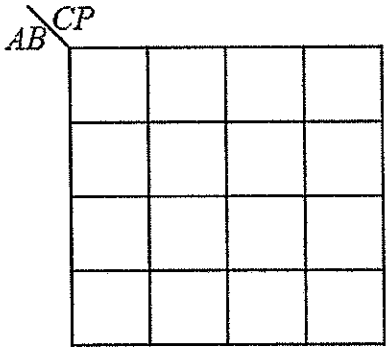
$T_A =$



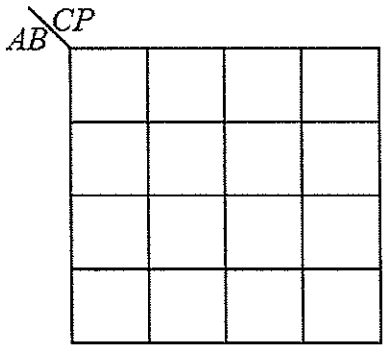
$J_B =$



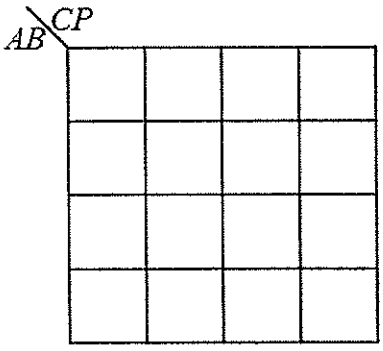
$K_B =$



$T_C =$



$J_P =$



$K_P =$

(1c) Draw the final circuit for the counter. (3 marks)

ANSWER:

QUESTION 2 (15 marks)

(2a) What is the MIPS instruction that is encoded by the hexadecimal number `0x2acaff85`? (2 marks)

ANSWER:

(2b) Encode the MIPS instruction "`xor $v0, $a1, $t9`", leaving your answer as an 8-hexadecimal digit number. (3 marks)

ANSWER:

(2c) Consider the following C function:

```
char *ToUpper(char *s1, char *s2);
```

It will copy the C string pointed to by **s1** to **s2** and convert all lower case letters to upper case ones. In other words, at the end of its execution, **s2** will be a copy of **s1** except that all lower case letters will be converted to upper case ones. All other characters remain unchanged. The function returns **s2** as the result. Write an *efficient* MIPS assembly routine for this function. Note that the ASCII code for 'A' (uppercase A) is 0x41, while the ASCII code for 'a' (lowercase a) is 0x61. (10 marks)

ANSWER:

QUESTION 3 (20 marks)

(3a) What is the number represented by the IEEE Standard 754 single precision floating point number `0xBDD40000`? Leave your answer in decimal scientific notation. (3 marks)

ANSWER:

(3b) What is the *smallest* and the *biggest* gap between two *adjacent* normalized IEEE Standard 754 single precision floating point numbers? (6 marks)

ANSWER:

(3c) Suppose an internal computation has resulted in the following floating point number:

Sign = 1

Exponent = 01100101_2

Fraction bits = $10001111111111111100110_2$

Guard bit = 1

Round bit = 0

Sticky bit = 1

What is the IEEE standard 754 single precision format number (leave your answer as a hexadecimal string) after (i) rounding to nearest, (ii) rounding to zero, and (iii) rounding to $+\infty$? (5 marks)

ANSWER:

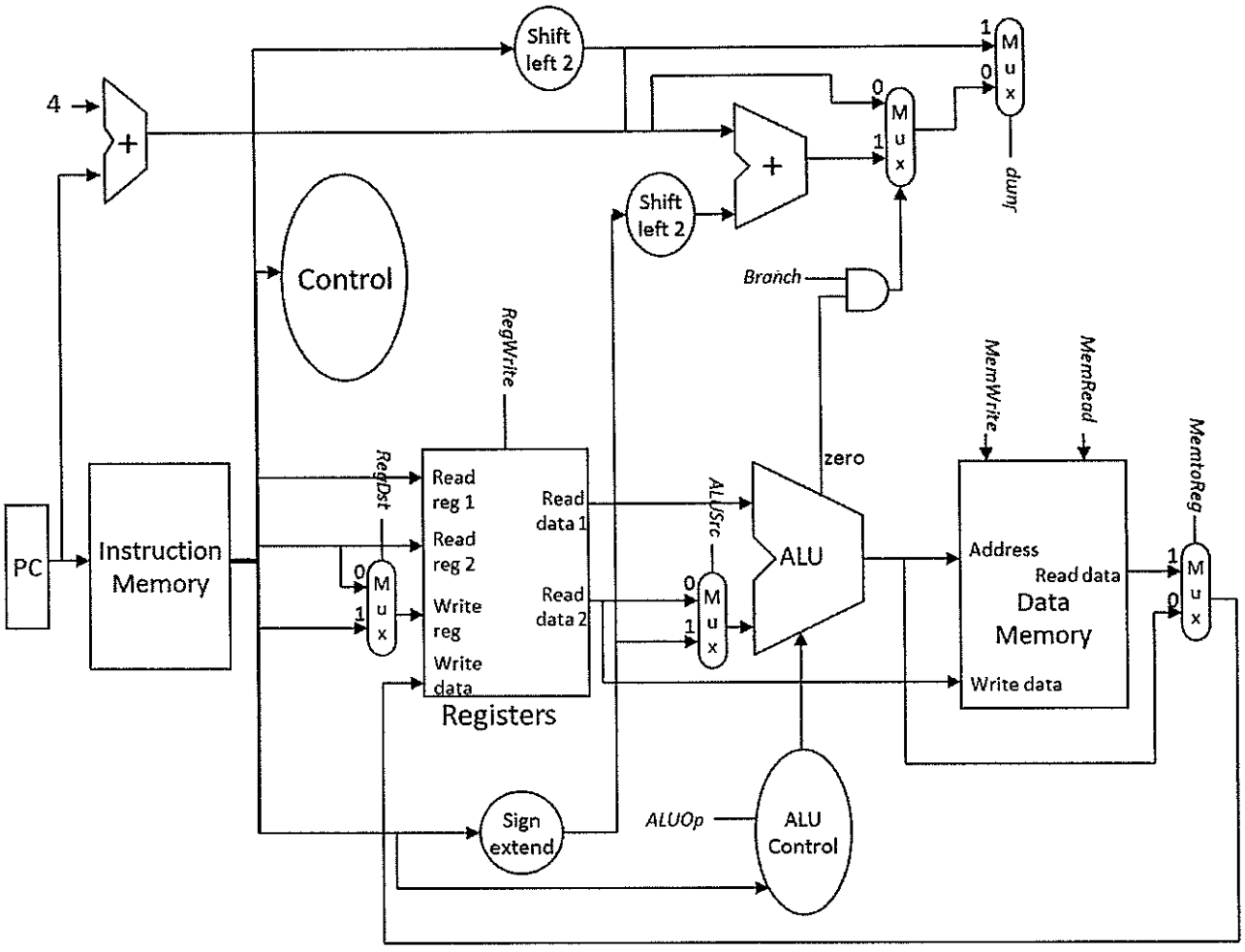
(3d) The associativity rule is not observed in floating point arithmetic. For example, in general, $(A + B) + C \neq A + (B + C)$ where A, B, and C are IEEE Standard 754 single precision floating point numbers. Give an example of A, B, and C that will illustrate this property. State clearly which rounding mode you are assuming in your example.

(6 marks)

ANSWER:

QUESTION 4 (25 marks)

(4a) The diagram below is a simplified version of the single cycle datapath.



Write down the control signals that would constitute the execution of the instruction:

```
addi  $s0, $t5, -1
```

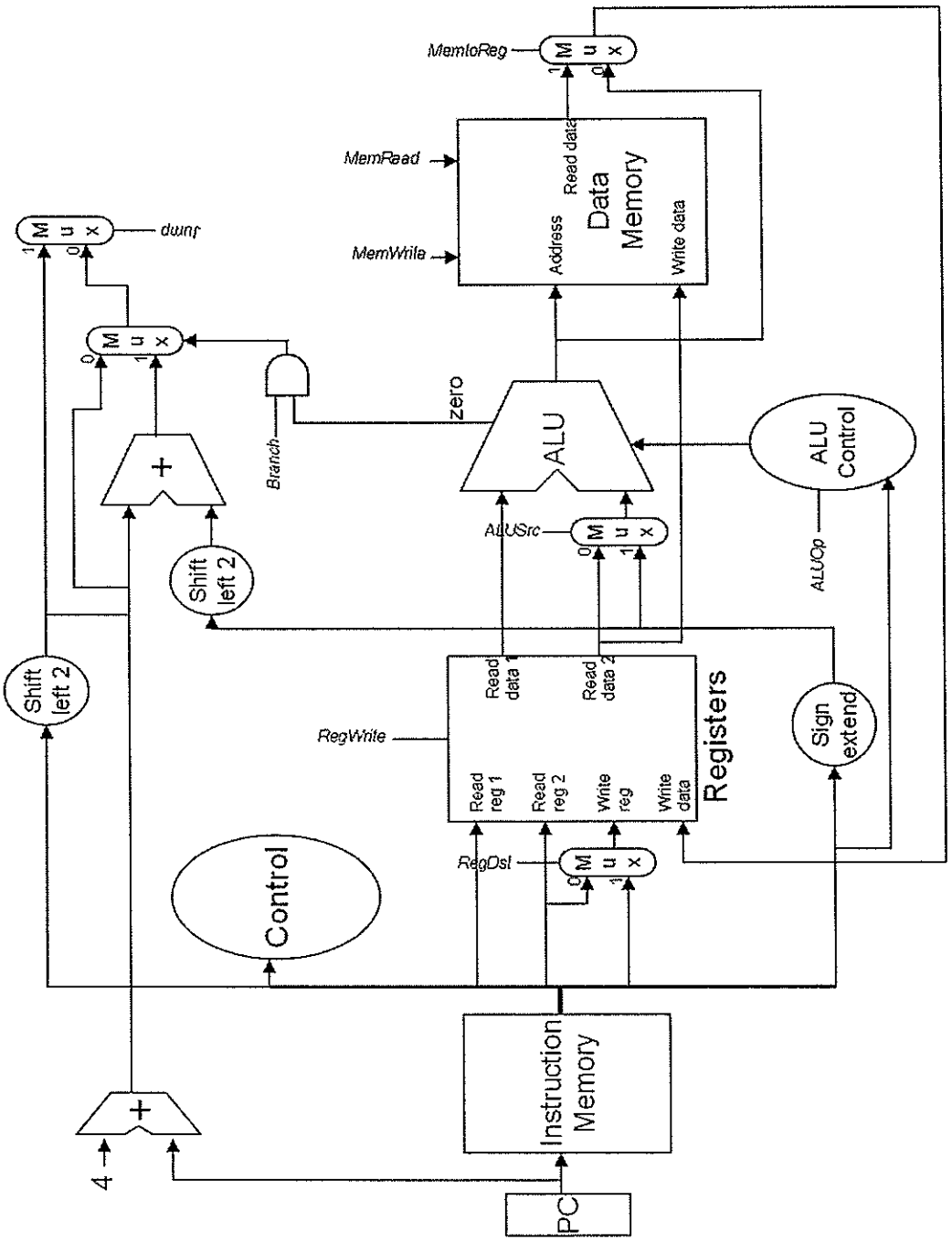
If any of the values are more than 1 bit, write down the hexadecimal value. You may also use 'X' for "don't care" or unknown values. However, assume that the contents of the both registers mentioned in the instruction above are zeroes. (5 marks)

ANSWER:

Signal	Value
Read reg 1	
Read reg 2	
Write reg	
RegDst	
RegWrite	
ALUSrc	
Branch	
Address	
ALUOp	
MemWrite	
MemRead	
MemtoReg	
Jump	

(4b) By drawing additional components and signal lines, including possibly new control signals (which you should define), show how support for the **`jal`** ("jump and link") instruction can be added. The **`jal`** instruction is a jump instruction that also writes PC+4 into register **`$31`**. Besides drawing new lines and control signals on the diagram, you should also identify any changes to existing control or data signals, and how the instruction is implemented. (10 marks)

ANSWER:



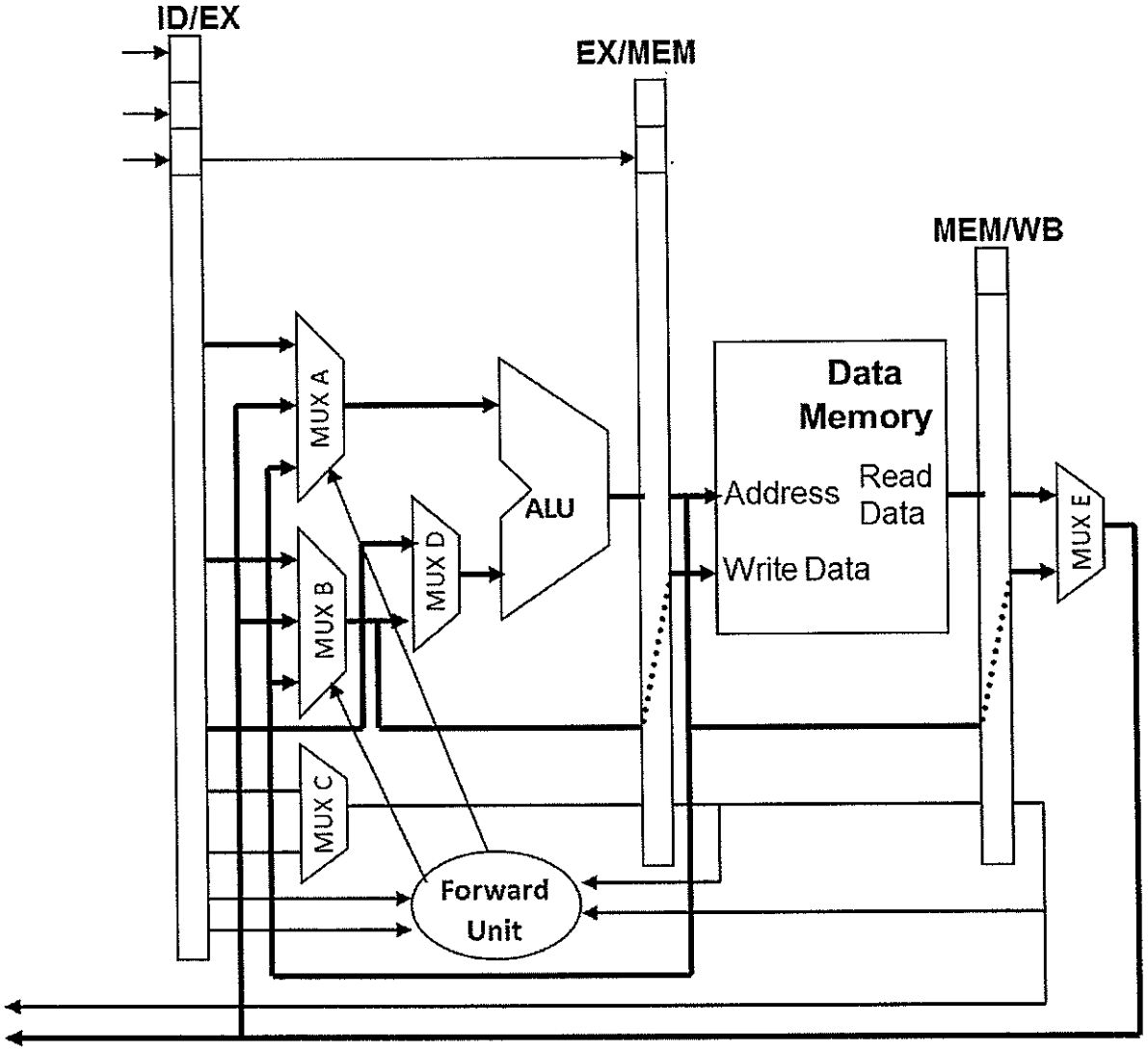
(Answer for (4b) continues here)

(4c) The diagram below is for the last three pipeline stages for a pipeline with forwarding facilities. Now consider the following two instructions:

```
add $1,$2,$3
sw  $1,4($3)
```

Describe how forwarding is done as these two instructions execute in these last three stages of the pipeline. You may want to annotate critical points in the diagram (say by drawing '①', '②', etc. on the diagram) so as to better explain the flow of data and the how the multiplexors implement that flow in each clock cycle. (5 marks)

ANSWER:



(Answer for (4c) continues here)

(4d) "Pipelining is one of the most important innovations in the architecture of processors." Explain the effects of pipelining as well as some of the drawbacks. (5 marks)

ANSWER:

QUESTION 5 (10 marks)

(5a) Show the block diagram implementation for a 2-way set associative, 2-word per block, 1Kbyte write-back cache on a byte-addressable processor. A 'word' here is 4 bytes. You should also show clearly how the 32-bit address is decomposed for addressing the cache. (5 marks)

ANSWER:

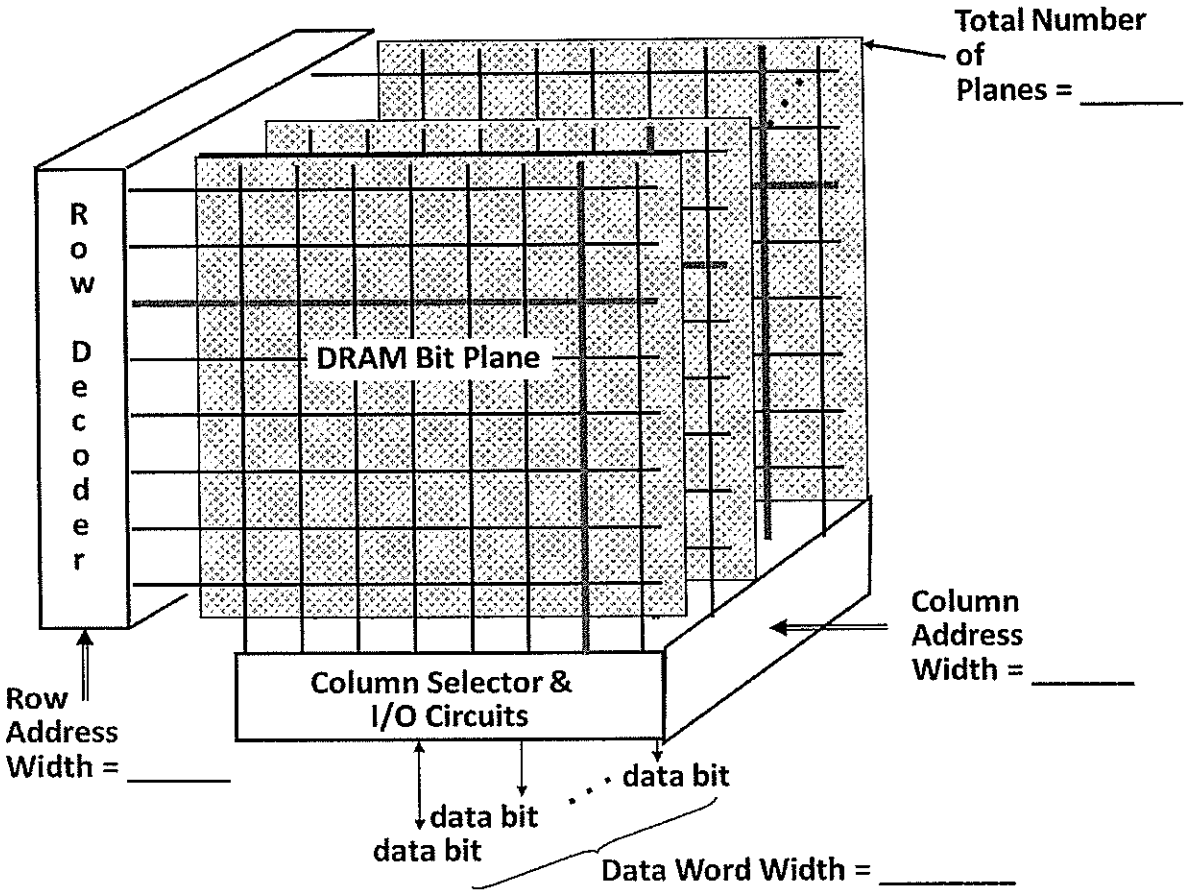
(5b) Propose a C/Java data structure for simulating this cache. You may assume that the type 'long' holds 4 bytes. (5 marks)

ANSWER:

QUESTION 6 (15 marks)

(6a) Using square DRAM bit planes of 512x512 bits, show how a 16 bit word DRAM organization would be like by filling in the details in the diagram below. (3 marks)

ANSWER:



(6b) What is the total size of the above implementation? (2 marks)

ANSWER:

(6c) Assume that a 32-bit address machine has a page size of 4,096 bytes, and a 4 entry TLB. The machine runs sillyOS that has a fixed page allocation algorithm: any virtual page is mapped to a physical page that has the virtual page number added with 100_{10}. The machine boots up and encounters four memory addresses:

```
0x10031023
0x3F291367
0x8888DEAD
0x0001671E
```

What would be the entries in the TLB (ignoring the dirty, ref, and permission bits) after the page faults for these four addresses have been successfully serviced. (5 marks)

ANSWER:

Valid Bit	Virtual Page Number	Physical Page Number

(6d) Why is the page allocation algorithm of sillyOS a very bad idea? (5 marks)

ANSWER:

=== END OF PAPER ===