

CONFIDENTIAL



UNIVERSITI TUN HUSSEIN ONN MALAYSIA

**FINAL EXAMINATION
SEMESTER I
SESSION 2017/2018**

TERBUKA

COURSE NAME : COMPUTER SYSTEMS
ENGINEERING

COURSE CODE : BEC41603

PROGRAMME : BEJ

EXAMINATION DATE : DECEMBER 2017 / JANUARY 2018

DURATION : 2 HOURS AND 30 MINUTES

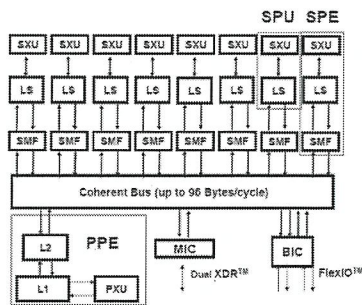
INSTRUCTION : ANSWER ALL QUESTIONS

THIS QUESTION PAPER CONSISTS OF EIGHT (8) PAGES

CONFIDENTIAL

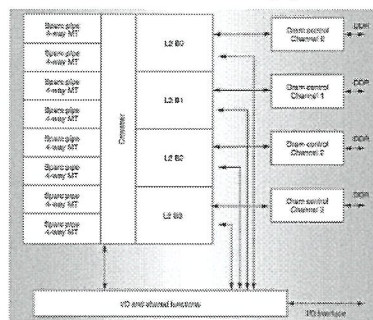
- Q1** (a) Differentiate between pipeline and superscalar architecture in processors using appropriate diagrams. (6 marks)
- (b) Describe a common problem for pipeline and superscalar processor. (2 marks)
- (c) Compare between in-order issue in-order completion and out-of-order issue out-of-order completion in superscalar processor in terms of hardware complexity and throughput. (4 marks)

Q2 Given the diagrams shown in **Figure Q2**, select the correct diagram for the following architecture and justify your answer:



Note: PPE – Power Processing Element, SPE – Synergistic Processing Element

Architecture (a)

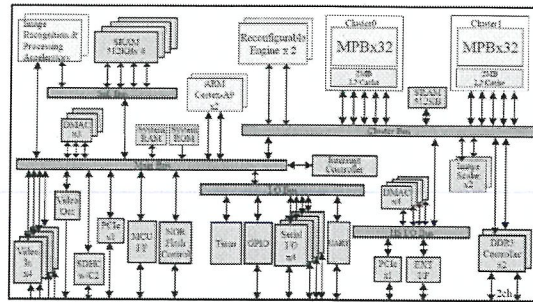


Note: SPARC – Scalable Processor Architecture

Architecture (b)

Figure Q2

TERBUKA



Note: MCU – Microcontroller Unit, MPB – Media Processing Engine Block

Architecture (c)

Figure Q2 (cont.)

- (i) Homogeneous architecture (2 marks)
 - (ii) Heterogeneous architecture (2 marks)
 - (iii) Mixed/irregular architecture (2 marks)
- Q3**
- (a) Compare between Network on Chip (NoC) and bus architecture in terms of parallel processing support using appropriate diagrams. (4 marks)
 - (b) Discuss one limitations of mesh NoC topology. (2 marks)
 - (c) Differentiate between circuit switching and packet switching. (4 marks)
- Q4**
- (a) State three types of memory access mechanism in multicore processor. (3 marks)
 - (b) Explain the impact of cache coherency to the performance of multicore processors as the number of processing cores increases. (4 marks)

TERBUKA

- (c) Predict the states of memory location $0 \times 0fd4$ in cache A, 0×0430 in cache B, $0 \times bb20$ in cache C and $0 \times 03dc$ in cache D shown in **Figure Q4(c)** using MESI coherency protocol after following process (1) – (3) is executed:

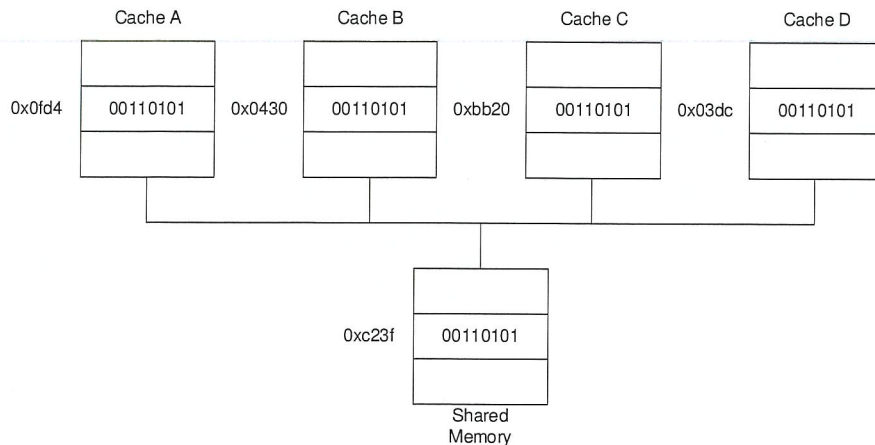


Figure Q4(c)

Conditions:

- (1) Initially, $0 \times 0fd4 = 00110101$, $0 \times 0430 = 00110101$, $0 \times BB20 = 00110101$, $0 \times 03dc = 00110101$.
- (2) Then, processor A modifies cache A address $0 \times 0fd4$ to 11111010 .
- (3) Next, processor A writes the updated data to the main memory.

(8 marks)

- Q5** (a) Based on the code snippet shown in **Figure Q5(a)**, determine the types of data dependence and its lines of code. Justify your answer.

```

1 | int temp = 0, b[10], a[10];
2 |
3 | for (i=0; i++; i<10) {
4 |     b[i] = temp;
5 |     a[i] = b[i]+5;
6 |     temp = a[i]+10;
7 | }
```

Figure Q5(a)

(6 marks)

TERBUKA

- (b) Assume that a computer has 16 cores that can be used to execute an application in parallel and 88% of application code is parallelizable.
- (i) Using Amdahl's Law, calculate the numbers of cores that are needed to achieve a speedup of 8. (3 marks)
- (ii) Determine whether it is possible to achieve a speedup of 6 using the same law for the application on the given computer. (4 mark)
- (c) Answer the following questions based on the code shown in **Figure Q5(c)**:

```
1 | int main ( )
2 | {
3 |     int i, n = 1000;
4 |     double s = 1.23, x[1000], y[1000];
5 |     for ( i = 0; i < n; i++ )
6 |     {
7 |         x[i] = ( double ) ( ( i + 1 ) % 17 );
8 |         y[i] = ( double ) ( ( i + 1 ) % 31 );
9 |     }
10 |     for ( i = 0; i < n; i++ )
11 |     {
12 |         x[i] = x[i] + s * y[i];
13 |     }
14 |     return 0;
15 | }
```

Figure Q5(c)

- (i) Determine the lines of code that can be executed in parallel. (2 marks)
- (ii) Write the OpenMP syntax to get the number of threads in the parallel region. (2 marks)
- (iii) Write the OpenMP syntax to set the number of threads in the parallel region to 8. (2 marks)
- (iv) Write the OpenMP syntax to get the number of each thread in the parallel region. (2 marks)

- (v) Write the OpenMP syntax to parallelize the *for loop* with private variable *i* and shared variable *s*.
(2 marks)
- (vi) Write the OpenMP syntax to get the number of processor cores available.
(2 marks)
- (vii) Write the OpenMP syntax to get the maximum number of threads that can be used.
(2 marks)

- Q6** (a) Based on the C++ code snippet shown in **Figure Q6(a)**, answer the following questions:

```
1 | int s = 0, N = 100, A[100];  
2 | #pragma omp parallel for  
3 | for (i=0; i<N; i++)  
4 | {  
5 |     A[i] = i + 1;  
6 |     s+ = A[i];  
7 | }
```

Figure Q6(a)

- (i) Decide whether data race condition happens in the code. If yes, write the specific lines of code and explain.
(3 marks)
- (ii) Give a reason whether the answer in Q6(a)(i) is considered as the critical section.
(2 marks)
- (iii) Write two suitable OpenMP directives to execute the critical section.
(4 marks)

TERBUKA

- (b) Given the C++ code snippets shown in **Figure Q6(b)**, answer the following questions:

```
1 | #pragma omp parallel for
2 | for (i=0; i<n; i++)
3 | {
4 |     y[i] = a * x[i] + y[i];
5 | }
6 | #pragma omp parallel for
7 | for (i=0; i<n; i++)
8 | {
9 |     z[i] = b + y[i];
10 | }
```

Figure Q6(b)

- (i) Determine whether barrier synchronization is needed at the end of the first parallel region (line 3 to 6). Justify your answer. (3 marks)
- (ii) State the OpenMP syntax to disable implicit barrier synchronization. (1 mark)
- Q7** (a) Differentiate between Central Processing Unit (CPU) and Graphic Processing Unit (GPU) architecture in terms of how computation is performed one by one. Use the following matrix multiplication $C[2] = A[2] \times B[2]$ as the computation example.

$$\begin{bmatrix} C11 & C12 \\ C21 & C22 \end{bmatrix} = \begin{bmatrix} A11 & A12 \\ A21 & A22 \end{bmatrix} \times \begin{bmatrix} B11 & B12 \\ B21 & B22 \end{bmatrix}$$

(4 marks)

- (b) GPU is suitable for SIMD type applications. Elaborate the performance effect when GPU is used to execute non-SIMD applications such as SISD and MIMD. (3 marks)

TERBUKA

- (c) GPU execution process involves the following steps shown in **Figure Q7(c)**. Rearrange the processes in the correct execution order.

Free device memory
Memory copy from host to device
Memory space allocation for device
Call kernel “multiply” for parallel execution in device
Memory copy from device to host

Figure Q7(c)

(5 marks)

- (d) List the GPU functions to be used for each process listed in **Figure Q7(c)** based on the correct execution order.

(5 marks)

- END OF QUESTIONS -