



Reg. No. : 

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

**Question Paper Code : 50403**

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2017  
Eighth Semester  
Computer Science and Engineering  
CS 6801 – MULTI-CORE ARCHITECTURES AND PROGRAMMING  
(Regulations 2013)

Time : Three Hours

Maximum : 100 Marks

Answer ALL questions

PART – A

(10×2=20 Marks)

1. Differentiate symmetric memory architecture and distributed memory architecture.
2. What are multiprocessor systems and give their advantages.
3. What are conditions under which a deadlock situation may arise ?
4. Define thread. Mention the use of swapping.
5. Define message queue.
6. What is termed as initial task region ?
7. List the restrictions to work sharing constructs.
8. Write the performance evaluation methods in distributed memory programming.
9. What is race condition ?
10. What are the features of distributed memory ?

PART – B

(5×16=80 Marks)

11. a) Explain in detail, the SIMD and MIMD systems. Discuss briefly the performance issues of multi-core processors.

(OR)

- b) Define Cache Coherence Problem. What are the 2 main approaches to cache coherence ? Describe working of snooping cache coherence and explain directory based coherence.



12. a) Explain the various approaches to Parallel Programming.

(OR)

b) What is a data race ? What are the tools used for detecting data races ? How to avoid data races ?

13. a) Illustrate an OpenMP execution model with an example.

(OR)

b) Explain in detail about the handling loops in parallel operations.

14. a) What is MPI ? Write a program "hello, world" that makes some use of MPI. How to compile and execute MPI programs ?

(OR)

b) Differentiate collective and point-to-point communication and draw the architecture for tree structured communication.

15. a) What does the n-body problem do ? Give the pseudocode for serial n-body solver and for computing n-body forces.

(OR)

b) How will you parallelize the reduced solver using OpenMP ? How will you parallelize the reduced solver using OpenMP ?

---